
LOAD
REVISTA PARA USUARIOS DE
MSX
AÑO 1 Nº 7, ≠ 2,70 REP. ARGENTINA
CONCURSO
ULTIMO MES
PROGRAMA
COMERCIAL
PRESTAMOS Y
AMORTIZACIONES
LOGO PARA
APRENDER Y JUGAR
EL ARTE DE
GRAFICAR
NOVEDADES
EN SOFTWARE
DESNUDAMOS EL
CHIP DE SONIDO
TOSHIBA
STOP



A la **Talent MSX** nada le es imposible

El Club de Usuarios de MSX ya funciona en sus tres direcciones: CABILDO 2027 - 1º A ESMERALDA 320 - 5º y TUCUMAN 2044 - 1º - CAPITAL

Invitamos a los felices usuarios de la TALENT MSX al **curso gratuito** de introducción al fabuloso mundo de MSX.

Participe del Club de Usuarios de MSX y encuéntrese con sus amigos que también tienen la TALENT MSX, e intercambiará programas, datos y chimentos.

Podrá probar todos los accesorios de la línea MSX, ¡¡desde disketteras hasta robots!!

Podrá ver y leer todo lo que le interese sobre la norma MSX: catálogos, libros y revistas de todo el mundo. Todo con la seguridad, respaldo y seriedad que sólo TALENT puede brindarle.

¡Para inscribirse, no olvide traer su factura de compra!

Club Talent MSX

MSX es marca registrada de MICROSOFT CORPORATION.



AÑO 1 Nº 7



Director General
Ernesto del Castillo

Director Editorial
Cristian Pusso

Director Periodístico
Fernando Flores

Director Financiero
Javier Campos Malbrán

Secretario de Redacción
Ariel Testori

Redacción
Eduardo Mombello
Andrea Sabin Paz

Arte y Diagramación
Fernando Amengual
Tamara Migelson

Departamento de Avisos
Oscar Devoto
Nelso Capello

Departamento de Publicidad
Guillermo González Aldalur

Servicios Fotográficos
Víctor Grubicy
Comesaña

Load Revista para usuarios de MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5º Piso, (1017) Buenos Aires. Tel. 46-2886 y 49-7130. Registro Nacional de la Propiedad Intelectual: E.T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual.

Precio de este ejemplar: A 2,70
Impresión: Calcotam. Fotocromo tapa: Columbia. Fotocomposición: Interamericana Gráfica
Los ejemplares atrasados se venderán al precio del último número en circulación.

Distribuidor en Capital: Martino, Juan de Garay 358, P.B. Capital.
Distribuidor interior: DGP: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800.

CORREO ARGENTINO SA CENTRAL | FRANQUEO PAGO CONCESION Nº 2538 | TARIFA REDUCIDA CONCESION Nº 1304

Sumario

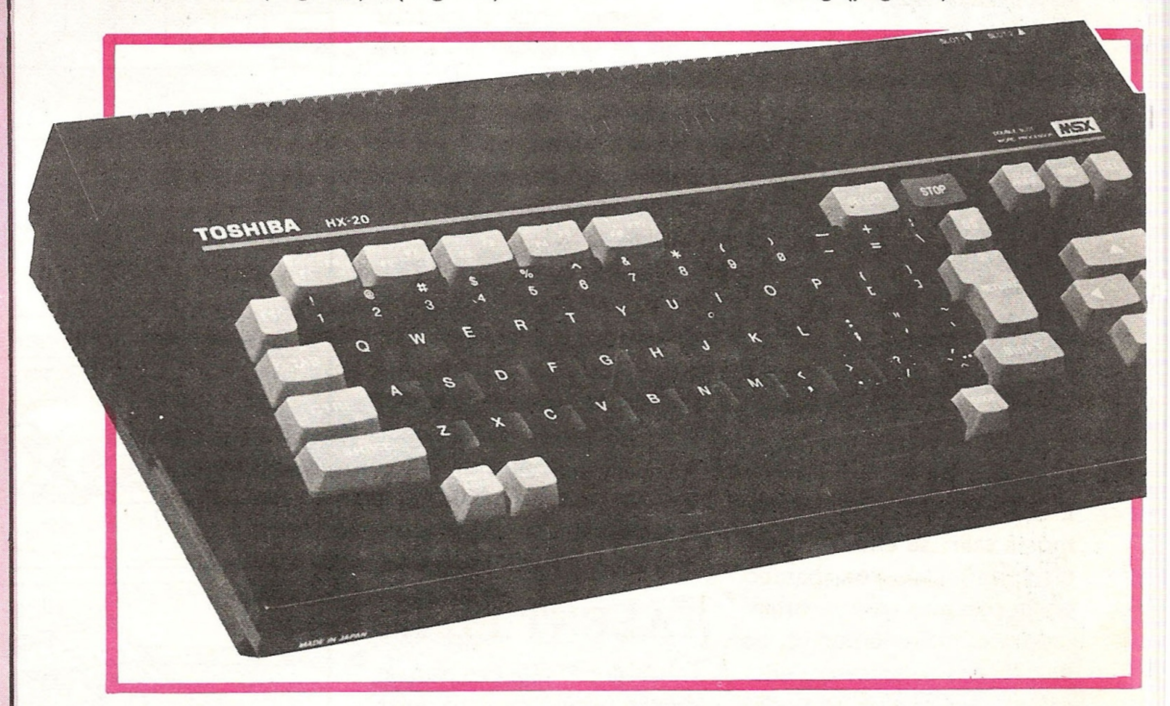
MAYOR COMPATIBILIDAD

La aparición de una nueva computadora con norma MSX en el mercado argentino es un acontecimiento significativo como se puede apreciar en la nota que publicamos en este número. Viene a reafirmar las esperanzas y esfuerzos de todos aquellos que se inclinaban por la compatibilidad de equipos y sistemas.

Este hecho sirve para que las dificultades en la interconexión de computadoras y periféricos de diferente origen se vean reducidas. En todo caso es un paso más para que las posibilidades del usuario se amplíen y la computadora se convierta en la "herramienta" que debe ser.

De esta manera avanzamos un tramo más en ese arduo camino que lleva a alcanzar la madurez dentro del ámbito tecnológico de nuestra informática. La "standarización" que se está logrando con la norma MSX es un hecho y los que manejan el mercado así parecen entenderlo.

LOS EDITORES





ASCII EN LUGAR DE MICROSOFT

Avalado por la venta de más de un millón de computadoras de norma MSX solamente en JAPÓN durante el año 1985, este sistema standard se impone cada día más.

Aunque algunos fabricantes se empecinan en cambiar ciertos aspectos de las computadoras que producen, con los fines que ya conocemos, los creadores del MSX se proponen todo lo contrario, y más.

Esto queda demostrado por el precontrato firmado por ASCII y MICROSOFT gracias al cual la primera ha tomado las riendas de este standard. No sólo se proponen lograr la compatibilidad entre todas las máquinas que se han dado en llamar de norma "MSX" sino avanzar sobre la compatibilidad entre estas y las denominadas PC.

Y esto es un hecho. De la mano de ASCII (ahora dueña de los caminos que tomará la norma) se están creando —inclusive por ellos mismos— grandes paquetes de software que harán saltar de su silla a más de uno.

Por otro lado, ASCII se propone, con agresivas campañas, plagar con la norma el mercado europeo y si es posible mundial.

Entre sus desarrollos próximos a salir, se encuentra un digitalizador de voz, basado en un chip muy barato, orientado a complementar varios desarrollos de software educativo. Así, el alumno podrá hablar directamente con la computadora sin intermediarios. Su funcionamiento es similar al digitalizador de imágenes que, aparte, ya tiene varias versiones y fabricantes japoneses, aunque aquí todavía no es común verlos.

Por otro lado, la firma KONAMI (japonesa) anunció que está trabajando en un nuevo desarrollo de software (un juego) que incluye impresionantes pantallas utilizando otro de los chips engendrados por los japoneses. Recordemos que esta firma es una de las más importantes creadoras de juegos para esta norma.

COMPACT DISC

Los orientales continúan ampliando las posibilidades de las MSX incorporándoles accesorios potentes.

Esta vez, la empresa PIONEER decidió agrandar esta expansión de las computadoras de norma estandar, conectando un ordenador con un video.

Pero esto no es lo insólito, sino que el tratamiento de imágenes se efectuara con un Compact Disc.

Esta firma de electrónica ha fabricado una computadora con el P-BASIC, aplicado para el manejo del video.

Los japoneses han conseguido la vanguardia al mezclar las computadoras con las máquinas de video, y el hecho de haber incorporado este nuevo sistema del Compact a los procesamientos de imágenes, marca que todo ya es posible.

Inclusive, los Compact Disc se transformarán en una unidad de almacenamiento masivo, CP-ROM, con mucha más capacidad de guardar datos que los disketes, dando fin a las mediciones en kbytes, porque se hablará de la capacidad en Megabytes.

TALENT PLAN

Talent creó un nuevo plan de ahorro previo para no dejar a nadie sin la posibilidad de contar con una MSX. Son numerosas las ventajas. Anticipar las cuotas nos evita pagar los gastos administrativos. Las cuotas pagadas se actualizan automáticamente. Al pagar cuotas anticipadas también congela definitivamente el porcentaje pagado de la MSX y/o periféricos al igual que todas las cuotas abonadas.

Si decidimos desistir del PLAN, lo abonado se restituye actualizado al finalizar el contrato, con una disminución del 2% (renuncia) o del 4% (rescisión).

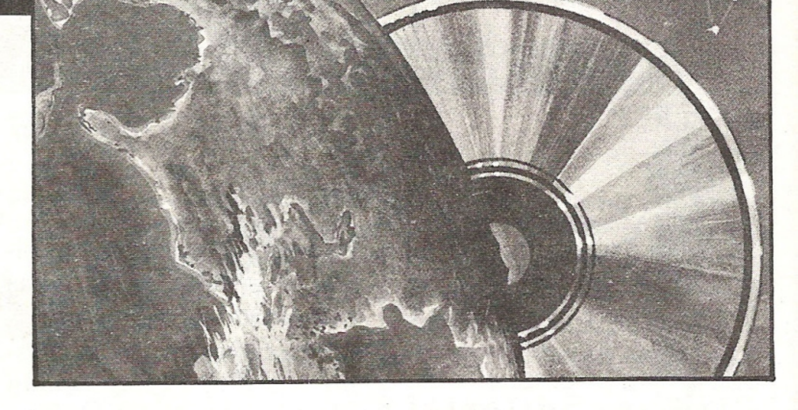
Todos los meses se sortea una computadora aparte de la licitación.

Este nuevo plan en realidad son tres. Es decir, hay tres planes, cada uno con diferentes opciones, pero la forma de ahorro es la misma.

Plan 1: computadora MSX DPC 200 (alternativa Unidad de Disco Flexible Talent MSX DPF-550 o 555 sin diferencia de precio).

Plan 2: computadora Talent MSX DPC200 y unidad de disco flexible 550 o 555 (alternativa reemplazar computadora o unidad de disco flexible por monitor TV Talent 14'' sin control y la diferencia para cancelar cuotas).

Plan 3: computadora personal MSX DPC 200, unidad de disco DPF 550 ó 555 y monitor TV Talent 14'' sin control.



CONCURSO

DE PROGRAMAS

auspiciado por TELEMATICA S.A. que proveerá lo siguientes Premios:

PRIMER PREMIO

UN PERIFERICO (a elección entre un monitor, una diskettera y una impresora).

UNA BECA para trabajar en el Departamento de Investigación y Desarrollo de Telemática S.A.

SEGUNDO PREMIO

UN PERIFERICO (a elección entre un monitor, una diskettera y una impresora).

En caso de que el ganador no pueda utilizar la beca será ofrecida a quien obtenga el segundo premio, y si éste tampoco pudiera aprovecharla se otorgará a alguno de los participantes del certamen que se hubiera destacado.

ESPECIAL Entre los programas recibidos, algunos de ellos podrán ser editados por Prosoft, reconociéndose los derechos de autor.

JUEGOS:

Tema: DE INTELIGENCIA

Los juegos que nos inspiren podrán ser "El estanciero" o temas originales que sigan la línea.

Tema: TRUCO

Premiaremos al programador que logre la mejor versión de este tradicional juego de salón.

EDUCATIVOS:

Tema: LIBRE

UTILITARIOS:

Tema: INTELIGENCIA ARTIFICIAL

Se trata de lograr un programa que realmente nos haga discutir sobre el límite entre la simulación y la inteligencia de la computadora.

Obviamente la única forma de lograr esto será siguiendo las reglas propuestas de la Inteligencia artificial. Por esto se considerarán para premiar esta categoría, además de las reglas detalladas más abajo, la capacidad de auto-aprendizaje del programa, el nivel de inferencia del mismo, la capacidad y modo de almacenamiento de su base de datos, y principalmente su analizador sintáctico dado que hasta el momento no se ha logrado uno que dé suficiente credibilidad de que estamos frente a un ser racional que entiende nuestro idioma y no frente a una máquina a la que debemos hablarle con verbos en infinitivo al mejor estilo Tarzán (sin menospreciar a este último)

PROFESIONAL O GESTION:

Tema: LIBRE

Dentro de este tema podrán figurar programas de las más diversas aplicaciones.

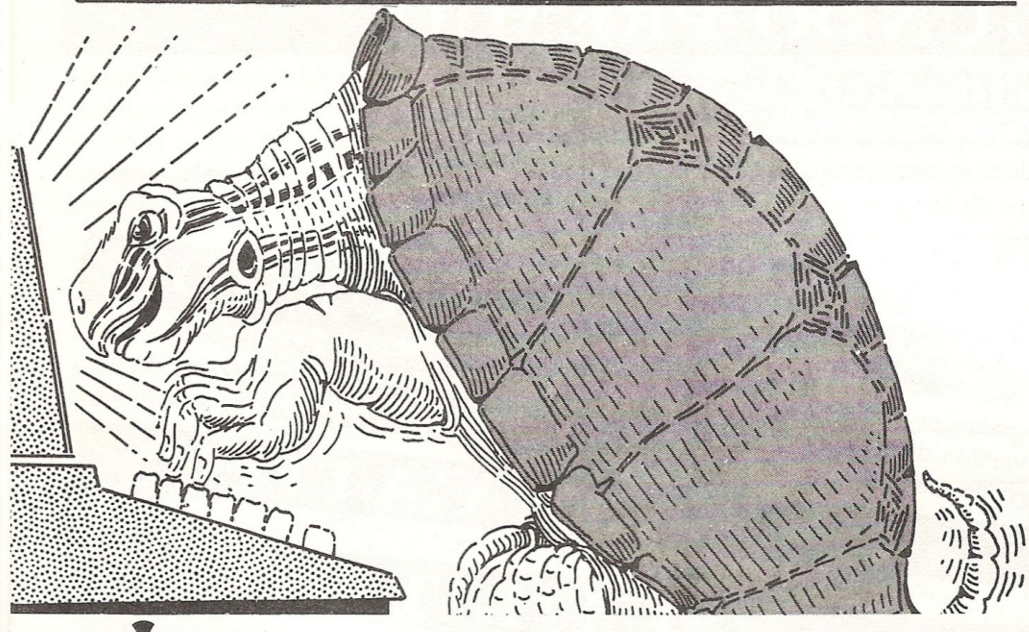
BASES: No sólo será indispensable que el programa enviado funcione correctamente sino que además debe cumplir con ciertas reglas.

- *Programación estructurada en bloques fácilmente diferenciables.*
- *Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de línea que los identifiquen).*
- *Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.*
- *Las variables y/o direcciones de memoria utilizados también se deben incluír en esta documentación.*
- *Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.*
- *Calidad y originalidad de gráficos, sonidos y pantallas de menú.*
- *El programa debe ajustarse a alguno de los temas propuestos más arriba. Esto es ELIMINATORIO.*

Los trabajos deberán enviarse antes del 30 de noviembre próximo (cierre del certámen) a: Paraná 720, piso 5°, (1017) Capital Federal.

LOGO PARA APRENDER Y JUGAR

Permite introducirnos facilmente en el mundo de las computadoras sin instrucciones complicadas, en castellano y con el mismo sentido que las palabras que usamos cotidianamente para expresarnos.



Se trata de un lenguaje de programación que nació en el Laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachusets a fines de 1960.

Debido al alto nivel de fracaso escolar en la asignatura de matemáticas, un equipo de investigadores de ese instituto, orientó la creación de un nuevo lenguaje, para solucionar el problema de los estudiantes.

El equipo estaba dirigido por un matemático y pedagogo llamado Seymor Papert. De alguna manera, Papert llevó al campo informático las ideas de su maestro, el psicólogo Jean Piaget. Seymor consiguió fijar los cimientos donde se apoyaría, luego, uno de los lenguajes de alto rendimiento para el usuario, además de ser de fácil comprensión.

Por su sencillez de manejo, permite ser adoptado por personas de todas las edades que se inicien en la programación.

Se buscó principalmente, un elemento didáctico con miras a utilizarlo para el aprendizaje y desarrollo de la capacidad intelectual de los alumnos.

Fue así como empezó a encasillarse al Logo como de uso casi exclusivo en la enseñanza, que permitía a los niños experimentar por sí mismos y desarrollar el conocimiento de una forma constructiva, ayudándolos a comprender ideas abstractas (como conceptos matemáticos), con el empleo de las computadoras. El Logo derivó del lenguaje Lisp, muy empleado para problemas de inteligencia artificial. Este último lenguaje es algo complicado de manejar, especialmente por la enorme cantidad de paréntesis que utiliza.

Aplicaciones y avances

MSX-Logo deja de ser de uso exclusivo para la educación. Gracias a su potencia, permite desarrollar cálculos complicados para sistemas expertos o trabajar directamente con la memoria de la computadora (como POKE, PEEK, etcétera en BASIC).

Una versión bastante potente de este lenguaje fue el TI-LOGO, empleado por las computadoras Texas Instruments. Las ventajas del MSX-Logo son, fundamentalmente, los comandos y operaciones en castellano, además de su optimización, satisfaciendo así las necesidades de cada vez más usuarios.

La facilidad de asimilación de este lenguaje por los chicos se debe, principalmente, a que el conjunto de órdenes que se pueden realizar ni bien son cargadas en la computadora son, al mismo tiempo, palabras con sentido para la computadora y para el lenguaje natural del usuario. A diferencia del BASIC, no hace falta contar con un diccionario inglés-castellano para traducir las palabras que se deben utilizar.

Logo se aproxima bastante a una charla entre el usuario y la computadora, donde el primero enseña a la computadora a realizar algunos procedimientos como dibujar un cuadrado, tocar una melodía, etcétera.

Crear gráficos

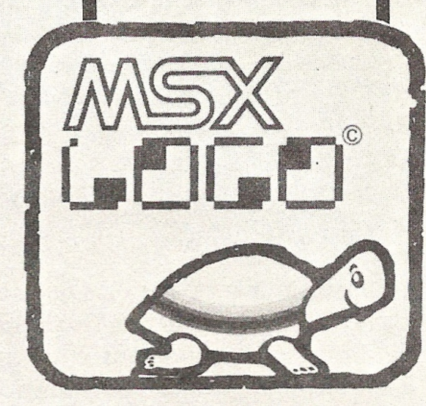
Logo permite crear interesantes gráficos en pantalla. Para saber cuál es nuestra ubicación en la pantalla, contamos con un "puntero". En realidad este "puntero" se representa con la figura de una tortuga. Pero mejor aún, se puede cambiar la representación del "puntero" por otras figuras ya definidas como una locomotora, un gato, un perro, un corazón y otras más. O bien, definir nosotros mismos el dibujo que más nos agrade.

Podemos darle órdenes a cada "puntero" o tortuga, como ADELANTE, ATRAS, o girar un ángulo determinado hacia la DERECHA o a la IZQUIERDA.

El recorrido de sus movimientos puede quedar marcado con distintos colores, a gusto del usuario. Pero por supuesto, también se puede desplazar la tortuga por la pantalla sin dejar marcadas sus huellas con la instrucción SINPLUMA.

Estas sentencias que permiten graficar en pantalla son simples de utilizar, como todas las instrucciones contenidas en MSX-Logo. Como dijimos, más bien parece un diálogo entre el usuario y la computadora, que termina con la realización, por parte de la máquina, de las tareas especificadas por el programador. Apoyado por el manual adjunto al car-

tucho MSX-Logo, el usuario puede experimentar sus propios inventos, como crear figuras geométricas de diversos tamaños. De esta manera, el futuro programador comienza a comprender la idea abstracta de "variable", palabra muy empleada en computación.



Manejo del lenguaje

Las instrucciones que forman al Logo pueden ser ejecutadas directamente. Esto permite probar el funcionamiento de las distintas instrucciones aisladas del programa total. Es decir, podemos ir probando instrucción por instrucción, con el fin de utilizarlas adecuadamente en un programa.

En este lenguaje se le enseña a la computadora a realizar otras tareas además de las ya implementadas en el lenguaje por sus fabricantes. Por ejemplo, se le dice cada paso que debe ejecutar para dibujar una silla.

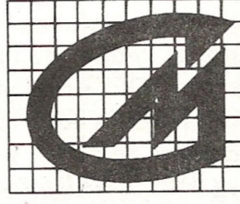
Estas tareas especificadas deben estar encerradas entre las palabras claves "PARA" y "FIN".

De esta manera, la computadora realizará todas las operaciones declaradas por el usuario que se encuentren entre estas dos palabras claves.

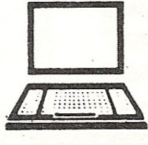
Con este procedimiento, es posible incorporar al lenguaje sentencias definidas por el usuario sin tener que ser expertos programadores.

A esta forma de programar se la denomina "modular", porque se construyen diferentes módulos, cada uno con una tarea específica.

En la figura 1 vemos un ejemplo de cómo se definen las sentencias. Para los que han trabajado con otros lenguajes, como Pascal, y tienen cierta experiencia en ellos, les vamos a mostrar por qué Logo es mucho más que un lenguaje pa-





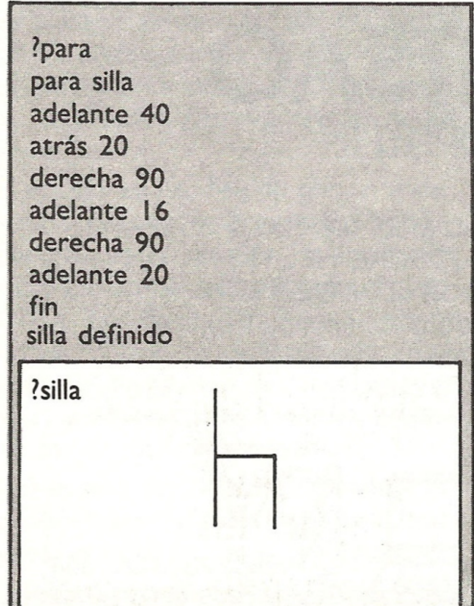


ra chicos. Una característica sobresaliente de Logo es que permite trabajar con "recursividad" (método de programación muy utilizado por los compiladores Pascal).
La definición de "recursividad" no es fácil de explicar con palabras. Se trata de un procedimiento que define una función en sí misma.
Esto, seguramente, para los que se inician en el área informática, no se entiende. Por eso, proponemos en la figura 2 un ejemplo. Como podemos notar, al final de la definición del procedimiento, éste se llama a sí mismo. A esto se le dice "recursivo".
La recursividad bien empleada facilita ampliamente la programación de sistemas complejos, porque permite expresar un problema como una versión reducida de sí mismo.
La interpretación de los listados de este lenguaje es sencilla (como se puede ver en los ejemplos dados) y permite encontrar los errores de programación rápidamente.
Esta es otra ventaja por la cual los pequeños programadores son atraídos.

Cálculos matemáticos

En MSX-Logo no se excluyeron las operaciones matemáticas tradicionales de suma, resta, multiplicación y división. También las operaciones trigonométricas de seno, coseno, arcotangente, etcétera, utilizadas por los usuarios más avanzados, se encuentran implantadas. Con la sentencia REDONDEO se obtiene la parte entera de un número, y con las sentencias AZAR o REAZAR se escogen números aleatorios.

Figura 1



Para operar con las variables lógicas, las sentencias NO, O e Y son las equivalentes en BASIC al NOT, OR y AND. Es decir, Logo contiene las operaciones del BASIC, más las operaciones trigonométricas, pero no en inglés.

Entrada y salida de datos.

Los datos pueden ser impresos en papel. Para esto se utiliza la instrucción CONIMPRESORA.
Para almacenarlos, se puede emplear cassettes o discos. En cada caso, las sentencias de "guardar" y "cargar" son diferentes. Pero ninguna es complicada de usar.

Código de máquina.

Los programadores más expertos, también podrán manipular directamente la memoria, creando programas en código de máquina. Estas son algunas de las sentencias:

- EXAMINAR: lee el dato situado en una posición determinada de la memoria. (Como PEEK en BASIC)
- DEPOSITAR: escribe un dato en una posición determinada de la memoria. (Como POKE en BASIC)
- DENTRO: lee un dato de los pórticos de entrada o salida. (Como IN en BASIC)
- LLAMAR: inicia la ejecución de un procedimiento escrito en código de máquina.

En fin, ésta es una muestra de las posibles sentencias a ejecutar para trabajar con código de máquina.

Definición de SPRITES y sonido

Para fabricar figuras, o sea SPRITES, se debe entrar al "editor de figuras" con la instrucción EDFIG n, donde n es el número de la figura a editar. Simplemente con las teclas de cursor y la barra de espacio se definen las figuras. Para salir del modo editor se presiona la tecla ESC.
Las figuras diseñadas permiten ser utilizadas para graficar nuestros programas. Y sobre el sonido, MSX-Logo permite entrar al mundo de la melodía, con creaciones propias. Se pueden crear melodías a tres voces que suenan simultáneamente.

Manual

El cartucho, como dijimos antes, viene acompañado por un manual con las explicaciones básicas sobre la forma de utilizar este lenguaje.
Comienza con la introducción al Logo. El primer capítulo es para familiarizarnos con las MSX. En esta sección, se le enseña al usuario cómo conectar correctamente el equipo para poder manejarlo. Aclara cómo se usan los discos y los cassettes, y explica las funciones de las teclas especiales como SHIFT, GRAPH, CODE y BS. En síntesis, es una introducción al manejo de una computadora a través del Logo.
Este manual trae una lista con las instrucciones con su correspondiente explicación rápida.
También están incluidas las explicaciones de los mensajes de error que aparecen en pantalla.
Al final de este compendio, encontraremos información sobre los códigos de colores, las figuras de las tortugas ya definidas, la tabla de caracteres, las frecuencias de tonos y un listado comparativo de las sentencias para las versiones de LOGO: USA, MSX y TI 99/4A.
Lógicamente, este manual está explicado en castellano, pero también trae una lista de sentencias Logo, traducidas del español al inglés y viceversa.
Resumiendo, estamos frente a una herramienta imprescindible para los que se inician enla programación y sobre todo para los más pequeños.

• Telemática S.A., licenciataria exclusiva de Logo Computer System Inc., para el lenguaje MSX-LOGO con adaptación de primitivas y redacción del Manual por los Ingenieros Hilario Fernández Long y Horacio Reggini.

Figura 2

```
?para
para polígono: ángulo
adelante 25 derecha: ángulo
hacer "ángulo": ángulo + 2
polígono: ángulo
fin

polígono definido

?polígono 45
```

PORQUE LA COMPUTACION ES EL FUTURO



MICROMATICA srl.

LOS PROFESIONALES DE LA COMPUTACION

ENTREGA INMEDIATA

DPC 200

SOFTWARE

- JUEGOS
- UTILITARIOS
- EDUCATIVOS
- A MEDIDA

HARDWARE

- EQUIPOS
- DISKETTERAS
- ACCESORIOS
- IMPRESORAS

BIBLIOGRAFIA - CURSOS (NIÑOS, ADULTOS, PROFESIONALES)

Logo - Basic - Introducción a la Informática - Inteligencia Artificial - Cobol - Fortram - Pascal - Base de Datos - Procesamiento de la palabra - Planilla Electrónica

DISTRIBUIDORES OFICIALES

Talent MSX

SPECTRAVIDEO

SVI

AV. PUEYRREDON 1135 (1118) Tel.: 961-5578

Suc. San Martín: Mitre 4044 San Martín (1650) TE: 752-6241

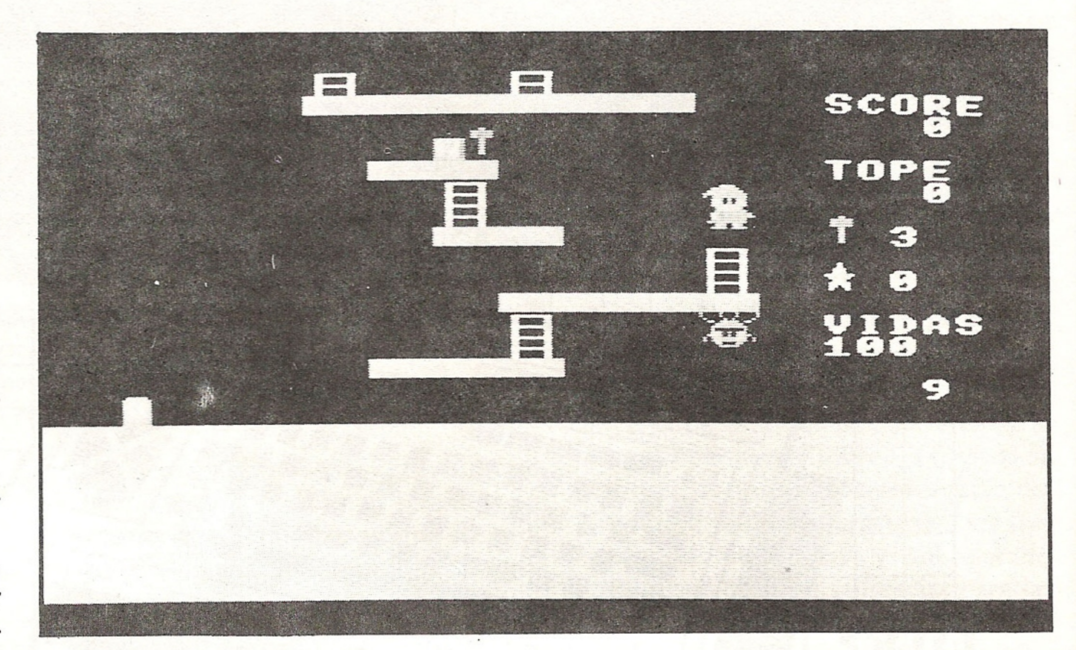


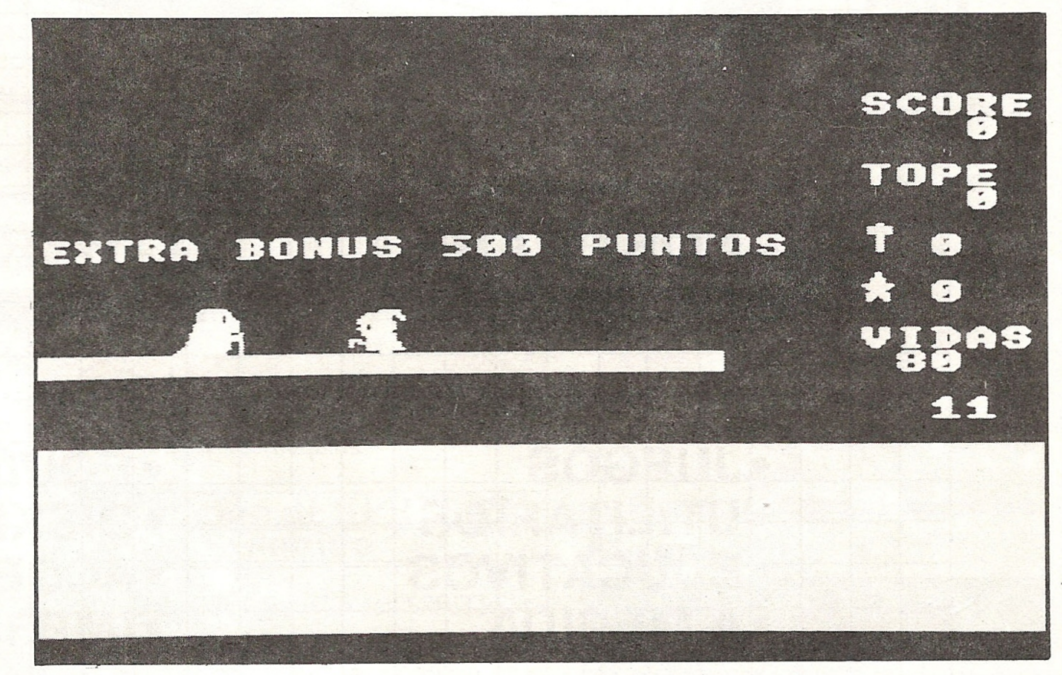
LAS ESCALERAS DEL DUENDE

Este es un divertido entretenimiento donde el personaje, un duende llamado PICKY, deberá subir y bajar escaleras o dejarse caer algunos pisos con el fin de encontrar cajas misteriosas.
Allí encontraremos diferentes objetos. Cada uno tiene un valor diferente.
Pero las cosas se complican cuando el duende es perseguido por una simpática abeja.
Al picar la abeja al duende, no lo mata, pero ya veremos qué pasa si sucede varias veces.
El programa tiene una pantalla principal. Para salir de ella tenemos que presionar la barra de espacio. Inmediatamente veremos cambiar el tamaño del sprite y se modificará la pantalla.
Si conseguimos mantenernos jugando por bastante tiempo, obtendremos como premio un bono extra de 500 puntos y pasar de nivel.
En cada nivel se incrementa la velocidad, transformando al juego con un clima tenso y desesperante.
Como vemos se trata de un entretenimiento bastante completo.
Pero hay que tener paciencia para copiarlo.
Para ayudarnos en la corrección del listado (si es que cometemos algún error), hemos definido la tecla de FUNCION 1 con: "SCREEN 1" para volver a obtener los caracteres comunes: y FUNCION 2 como: "WIDTH 29" para ampliar la ventana de la pantalla.
Si se brekea el programa (y si ya se redefinieron los caracteres), encontraremos imposible tratar de deducir el mensaje de error.
Para esto conviene presionar F1 y tener así caracteres legibles.

VARIABLES IMPORTANTES:

Z$: guarda el puntaje.
P: esta inicializado en 3. Se decrementa en 1 cada vez que perdamos una vida. Al llegar a 0 termina el juego.
T: tiempo de duración de cada nivel. Al llegar T a 360, dará el bono extra y pasaremos de nivel.





ESTRUCTURA DEL PROGRAMA:

10—120: inicializa las variables.
130—340: parte principal del programa.
350—370: dibuja piso.
380—430: distingue cual es el puntaje según el objeto recogido.
440—450: coloca el puntaje.
460—610: mueve el duende.
620—730: persecución y ataque de la abeja.
740—890: comienzo.
900—950: caer.
960—1000: chocar.
1010—1050: atacar.
1060—1130: pantalla para pasar de nivel.
1140—1220: perder.
1230—1280: fin del juego.
1290—1420: pantalla de presentación.
1430—1570: pantalla principal.
1580—1700: definición de caracteres.
1710—1760: lenguaje de máquina.
1770—1830: definición de los nuevos caracteres.
1840—1900: declaración de color.
1910—2130: definición de sprites.

```
10 SCREEN 1,2,0:WIDTH 29:DIM A$(
3),T$(3)
20 CLEAR 255,&HD7FF:COLOR 15,1,1
:KEY OFF
30 KEY 1,"SCREEN 1":KEY 2,"WIDTH
 29"
40 GOTO 1580
50 CLS:POKE &HF922,252:POKE&HF92
3,23:WIDTH 24
60 GOSUB 1290
70 A$(1)="           bc    bc
     bc    bc     "
80 A$(2)="           bc    bc
     bc    bc     "
90 A$(3)="     aaaaaaaaaaaaa  aa
aaaaaaaaaaaaaaaaa"
100 S=0:P=3:R=1:RT=0:L=100:X=3:Y
=8:V=0:PLAY"s0m2500t25518o5v15":H
=0:ST=0:W=0
110 GOSUB 1430
120 T=0:ZZ=0:TT=0:QT=0:M=0:N=0:M
E=3:M1=X*8:N1=Y*8:MN=22-R*2:MI=MN
:QQ=5
130 AA=STICK(0):ON AA GOTO 460,1
40,520,140,490,140,570
140 IK$=INKEY$:IK=INSTR(1,"¦zc "
,IK$):ON IK GOTO 150,620,740,870
150 M=M+SGN(M1-M)*4:N=N+SGN(N1-N
)*4:PUT SPRITE 3,(M,N),8,10
160 MI=MI-1:IF MI=0 THEN MI=INT(
MN):MN=MN+(MN>5)*.02:M1=X*8:N1=Y*
8:PUT SPRITE 3,(M,N),15,10
170 I=VPEEK(BASE(5)+X+Y*32):IF (
I>&H65) AND(I<&H69) THEN GOSUB 38
0
180 IF QT>0 THEN FOR I=1 TO 10:N
EXT
190 IF (ABS(M-X*8)<10) AND(ABS(N
-Y*8)<8) GOTO 960
200 QT=QT+1:IF QT<6-R-INT(T/80)
THEN 130:ELSE QT=0
210 IF TT=0 THEN GOSUB 290
220 IF T>360 GOTO 1060
230 TT=TT+1:LOCATE A*2+3,24:PRIN
TMID$(A$(TT)+A$(TT),C*2-1,B*2)
240 IF Y=0 GOTO 1140
250 Y=Y-1:FOR I=0 TO 2:VPOKE VDP
(5)*128+I*4,Y*8:NEXT I
260 IF TT>=3 THEN TT=0
270 Z$=RIGHT$("     "+STR$(L),3):
Q=12:GOSUB 440:IF L<=0 GOTO 1140
280 GOTO 130
290 Z=PEEK(RT+T)+ZZ:ZZ=INT(Z/20)
:C=Z+1-ZZ*20:Z=Z+ZZ
300 ZZ=INT(Z/6):A=Z+1-ZZ*6:Z=Z+Z
Z
310 ZZ=INT(Z/(9-A)):B=Z+2-ZZ*(9-
A):Z=Z+ZZ
320 T=T+1:Z$=RIGHT$("   "+STR$(T)
,4):Q=14:GOSUB 440
330 QQ=QQ-1:IF QQ=0 THEN GOSUB 3
50
340 RETURN
350 QQ=B:LOCATE A*2+1,18:PRINT"e
 ":LOCATE A*2+1,19:PRINT"aa"
360 LOCATE A*2+1,17:PRINTMID$("f
ffhgffffffhgfffhgfff",C,1)+" "
370 RETURN
380 PLAY"O6CFO5"
390 IF I=&H67 THEN LOCATE X,Y:PR
INT" o":LOCATE X+1,Y+1:PRINT"p":S
T=ST+3:Q=9:Z$="p"+STR$(ST)+" ":GO
SUB 440:RETURN
400 IF I=&H68 THEN LOCATE X,Y:PR
INT" k":LOCATE X+1,Y+1:PRINT"1":H
=H+3:Q=7:Z$="1"+STR$(H)+" ":GOSUB
 440:RETURN
410 S=S+100:Q=2:Z$=RIGHT$("    "+
STR$(S),4):GOSUB 440
420 LOCATE X,Y:PRINT" m":LOCATE
X+1,Y+1:PRINT"n"
430 L=L+5:Q=12:Z$=RIGHT$("    "+S
TR$(L),3):GOSUB 440:RETURN
440 '*****************************
**
450 FOR I=1 TO LEN(Z$): VPOKE BA
SE(5)+Q*32+24+I,ASC(MID$(Z$,I,1))
:NEXT I:RETURN
460 IF (VPEEK(BASE(5)+X+Y*32)<>&
H62) OR(Y<3)GOTO 150
470 Y=Y-3:V=0
480 PUT SPRITE 0,(X*8,Y*8),2+W*2
,7:PUT SPRITE 1,(X*8,Y*8),10,8:PU
T SPRITE 2,(X*8,Y*8),8,9:GOTO 150
490 IF (VPEEK(BASE(5)+X+Y*32+96)
<>&H62)OR(Y>12) GOTO 150
500 Y=Y+3:V=0
510 PUT SPRITE 0,(X*8,Y*8),2+W*2
,7:PUT SPRITE 1,(X*8,Y*8),10,8:PU
T SPRITE 2,(X*8,Y*8),8,9:GOTO 150
520 IF V<>1 THEN V=1:PUT SPRITE
0,(X*8,Y*8),2+W*2,4:PUT SPRITE 1,
(X*8,Y*8),10,5:PUT SPRITE 2,(X*8,
Y*8),8,6:FOR I=1 TO 9:NEXT:GOTO 1
50
530 IF X>19 GOTO 150 ELSE X=X+2
540 PUT SPRITE 0,(X*8,Y*8),2+W*2
,4:PUT SPRITE 1,(X*8,Y*8),10,5:PU
T SPRITE 2,(X*8,Y*8),8,6
550 IF ABS(VPEEK(BASE(5)+X+Y*32+
64)-&H60)>2 GOTO 900
560 GOTO 150
570 IF V<>-1 THEN V=-1:PUT SPRIT
E 0,(X*8,Y*8),2+W*2,1:PUT SPRITE
1,(X*8,Y*8),10,2:PUT SPRITE 2,(X*
8,Y*8),8,3:FOR I=1 TO 9:NEXT:GOTO
 150
580 IF X<3 GOTO 150:ELSE X=X-2
590 PUT SPRITE 0,(X*8,Y*8),2+W*2
,1:PUT SPRITE 1,(X*8,Y*8),10,2:PU
T SPRITE 2,(X*8,Y*8),8,3
600 IF ABS(VPEEK(BASE(5)+X+Y*32+
64)-&H60)>2 GOTO 900
610 GOTO 150
620 IF (V=0) OR(X+V*2<3)OR(X+V*2
).19 OR(ABS(VPEEK(BASE(5)+X+V*2+Y
*32+64)-&H60)>1)GOTO 150
630 H=H+W*(H.0):IF ((W=1)AND(H>0
))OR(VPEEK(BASE(5)+X+V*2+Y*32+64)
=&H60)GOTO 690
640 LOCATE X+V*2,Y:PRINT"  ":LOC
ATE X+V*2,Y+1:PRINT"ij"
650 PLAY"ce":FOR I=1 TO 2:NEXT I
660 LOCATE X+V*2,Y+1:PRINT"  ":L
OCATEX+V*2,Y+2:PRINT"''"
670 IF(ABS(Y*8-N)<10) AND(ABS(X*
8+V*16-M)<10)THEN GOSUB 1010
680 GOTO 150
```




```
690 LOCATE X+V*2,Y:PRINT"  ":LOC
ATE X+V*2,Y+1:PRINT"ij"
700 PLAY"ce":FOR I=1 TO 2:NEXT I
710 LOCATE X+V*2,Y+1:PRINT"  ":L
OCATE X+V*2,Y+2:PRINT"  "
720 Z$="1"+STR$(H)+" ":Q=7:GOSUB
 440
730 GOTO 150
740 '********************
750 IF(V=0) OR(X+V*2<3)OR(X+V*2)
>19 OR(ABS(VPEEK(BASE(5)+X+V*2+Y*
32+64)-&H60)<2)GOTO 150
760 ST=ST+W*(ST>0):IF((W=1)AND(S
T>0))OR(VPEEK(BASE(5)+X+V*2+Y*32+
64)+&H64)LIST 820
770 LOCATE X+V*2,Y:PRINT"  ":LOC
ATE X+V*2,Y+1:PRINT"ij"
780 PLAY"ce":FOR I=1 TO 2:NEXT I
790 LOCATE X+V*2,Y+1:PRINT"  ":L
OCATE X+V*2,Y+2:PRINT"dd"
800 IF(ABS(Y*8-N)<10) AND(ABS(X*
8+V*16-M)<10)THEN GOSUB 1010
810 GOTO 150
820 LOCATE X+V*2,Y:PRINT"  ":LOC
ATE X+V*2,Y+1:PRINT"ij"
830 PLAY"ce":FOR I=1 TO 2:NEXT I
840 LOCATE X+V*2,Y+1:PRINT"  ":L
OCATE X+V*2,Y+2:PRINT"aa"
850 Z$="p"+STR$(ST)+" ":Q=9:GOSU
B 440
860 GOTO 150
870 IF H+ST=0 THEN W=0:GOTO 150
880 W=-1*(W=0)
890 V=0:PUT SPRITE 0,(X*8,Y*8),2
+W*2,7:PUT SPRITE 1,(X*8,Y*8),10,
8:PUT SPRITE 2,(X*8,Y*8),8,9:GOTO
 150
900 '*********************
910 F=0
920 Y=Y+1:F=F+1:FOR I=0 TO 2:VPO
KE VDP(5)*128+4*I,Y*8:NEXT I
930 IF Y>16 GOTO 1140
940 IF ABS(VPEEK(BASE(5)+X+Y*32+
64)-&H60)>1 GOTO 920
950 GOTO 150
960 '*********************
970 PUT SPRITE 3,(M,N),13,10:PLA
Y"o2do5"
980 L=L-SGN(L)*20:Z$=RIGHT$("
"+STR$(L),3):Q=12:GOSUB 440
990 M=0:N=0:M1=X*8:N1=Y*8
1000 GOTO 200
1010 '*********************
1020 PUT SPRITE 3,(M,N),15,10:PL
AY"a":FOR I=1 TO 10:NEXT I
1030 ME=ME-1:IF ME=0 THEN ME=3:M
=0:N=0:M1=X*8:N1=Y*8:S=S+100
1040 S=S+10:Z$=RIGHT$("    "+SRT$
(S),4):Q=2:GOSUB 440
1050 RETURN
1060 PLAY"l8o5gr32gd4dgaa+r32a+g
4ga+o6cl16dddddddd"
1070 IF PLAY(0)<>0 GOTO 1070
1080 FOR I=1 TO 32:PRINT:NEXT I
1090 PUT SPRITE 0,(80,80),2,1:PU
T SPRITE 1,(80,80),10,2:PUT SPRIT
E 2,(80,80),8,3:PUT SPRITE 3,(0,0
),0,0
1100 LOCATE 5,10:PRINT"qr":LOCAT
E 5,11:PRINT"stu":PRINT"aaaaaaaa
aaaaaaaaaaa"
1110 LOCATE 1,7:PRINT"EXTRA BONU
S 500 PUNTOS"
1120 FOR I=1 TO 1000:NEXT I:S=S+
3000:R=R-(R<3):X=3:Y=8:V=0:W=0:T=
0
1130 PUT SPRITE 0,(24,64),2,7:PU
T SPRITE 1,(24,64),10,8:PUT SPRIT
E 2,(24,64),8,9:GOTO 110
1140 '*********************
1150 PLAY"o6ed+dc+co5ba+agr8dr8g
"
1160 IF P=0 GOTO 1230
1170 X1=0:FOR I=16 TO 10 STEP -1
:FOR J=21 TO 5 STEP -2
1180 IF (X1=0) AND (VPEEK(BASE(5
)+I*32+J)=&H61) THEN X1=J:Y1=I
1190 NEXT J:NEXT I:X=X1:Y=Y1-2:L
=100:Z$=" 100":Q=14:GOSUB 440
1200 V=0:W=0:PUT SPRITE 0,(X*8,Y
*8),2,7:PUT SPRITE 1,(X*8,Y*8),10
,8:PUT SPRITE 2,(X*8,Y*8),8,9
1210 P=P-1
1220 GOTO 130
1230 ' *********************
1240 FOR I=1 TO 12:LOCATE 0,24:P
RINT:NEXT
1250 PRINT"      TERMINO EL JUEGO
"
1260 FOR I=1 TO 17:LOCATE 0,24:P
RINT:NEXT
1270 IF HS<S THEN HS=S
1280 FOR I=1 TO 1000:NEXT I:GOTO
 60
1290 ' *********************
1300 FOR I=0 TO 5:PUT SPRITE I,(
0,0),0,0:NEXT I
1310 CLS:LOCATE 0,24:PRINT"  LOA
D MSX":PRINT
1320 PRINT"  aaa       a"
1330 PRINT"  a a o     a"
1340 PRINT"  aaa p aa a a a a"
1350 PRINT"  a   a a  aa  a a"
1360 PRINT"  a   a aa a a aaa"
1370 PRINT"                 a"
1380 PRINT"              aaaa"
1390 FOR I=1 TO 7:PRINT:PRINT:NE
XTI
1400 VDP(1)=VDP(1) OR 1:PUT SPRI
TEO,(24,88),2,7
1410 PUT SPRITE 1,(24,88),10,8:P
UT SPRITE 2,(24,88),8,9
1420 IF INKEY$=" " THEN VDP(1)=V
DP(1) AND 254:RETURN:ELSE GOTO 14
20
1430 ' *********************
1440 Z$="SCORE":Q=1: GOSUB 440
1450 Z$=RIGHT$("    "+STR$(S),4):
Q=2:GOSUB 440
1460 Z$="TOPE ":Q=4:GOSUB 440
1470 Z$=RIGHT$("    "+STR$(HS),4)
:Q=5:GOSUB 440
1480 Z$="k":Q=6:GOSUB 440:Z$="l"
+STR$(HN)+" ":Q=7:GOSUB 440
1490 Z$="o":Q=8:GOSUB 440:Z$="p"
+STR$(ST)+" ":Q=9:GOSUB 440
1500 Z$="VIDAS":Q=11:GOSUB 440:Z
$=RIGHT$("    "+STR$(L),3):Q=12:GO
SUB 440
1510 LOCATE 0,24:PRINT"   aaaaaa
aaaaaa"
1520 PRINT"        bc    bc"
1530 PRINT"        bc    bc"
1540 PRINT"      aaaaaaaaaaaaaa"
1550 PRINT"              bc":PRINT"
        bc":PRINT"      aaaaaaaa
"
1560 PRINT:PRINT:PRINT"
    aa"
1570 X=3:Y=11:RETURN
1580 ' *********************
1590 CLS:LOCATE 0,11:PRINT"Esper
a, se estan definiendo  los carac
teres"
1600 FOR I=0 TO 60:FOR J=0 TO 7:
Z=VPEEK(32*8+I*8+J):POKE&HD800+I*
8+J,ZOR(Z/2):POKE&HDB80+I*8+J,(15
+(J>4)*8)*16+1:NEXT J:NEXT I
1610 RESTORE 1770:FOR I=64 TO 85
:FOR J=0 TO 7:READ A$:POKE&HD800+
I*8+J,VAL("&h"+A$):NEXT J:NEXT I
1620 RESTORE 1840:FOR I=64 TO 85
:FOR J=0 TO 7:READ A$:POKE&HDB80+
I*8+J,VAL("&h"+A$):NEXT J:NEXT I
1630 I=0:RESTORE 1710
1640 READ A$:POKE&HDF00+I,VAL("&
h"+A$):IF A$<>"C9"THEN I=I+1:GOTO
 1640
1650 RESTORE 1910:FOR I=1 TO 11:
S$="":FOR J=0 TO 31:READ A$:S$=S$
+CHR$(VAL("&h"+A$)):NEXT J
1660 SPRITE$(I)=S$:NEXT I
1670 DEFUSR=&H7E:A=USR(0)
1680 DEFUSR=&HDF00:A=USR(0):GOTO
 50
1690 DEFUSR=&H7E:A=USR(0)
1700 DEFUSR=&HDF00:A=USR(0)
1710 ' *********************
1720 DATA 21,00,D8,11,00,01,01,0
0,03,CD,5C,00
1730 DATA 21,00,D8,11,00,09,01,0
0,03,CD,5C,00
1740 DATA 21,80,DB,11,00,21,01,0
0,03,CD,5C,00
1750 DATA 21,80,DB,11,00,29,01,0
0,03,CD,5C,00
1760 DATA 3E,77,21,00,30,01,00,0
8,CD,56,00,C9
1770 ' *********************
1780 DATA 00,00,40,04,65,FF,FF,0
0,00,00,40,04,65,FF,FF,00,1F,18,1
8,18,1F,18,18,18,F8,18,18,18,F8,1
8,18,18
1790 DATA 00,00,00,00,00,00,00,0
0,AB,8B,8B,93,83,93,83,FF,00,00,0
0,00,00,7E,C7,93,00,00,00,00,00,7
E,C7,93
1800 DATA 00,00,00,00,00,7E,C7,9
3,02,38,5D,BF,BF,FB,FB,3D,00,39,D
C,FE,EE,EE,EE,DC,00,00,00,00,18,7
E,7E,7E
1810 DATA 18,18,18,18,18,18,00,0
0,00,00,00,00,00,18,3C,3C,7E,7A,7
A,A7,3C,34,18,00,00,00,00,00,00,1
0,10,38
1820 DATA 38,FE,7C,38,7C,6C,44,0
0,01,03,03,03,03,07,07,07,F8,FC,F
4,F6,F7,FC,F8,F8,0F,0F,0F,1F,1F,3
F,7F,FF
1830 DATA FC,FC,FC,FC,FC,FC,F8,F
8,80,80,80,80,80,80,80,80
1840 ' *********************
1850 DATA 97,97,97,97,97,97,97,1
1,97,97,97,97,97,97,97,11,F1,91,9
1,91,F1,91,91,91,F1,91,91,91,F1,9
1,91,91
1860 DATA 11,11,11,11,11,11,11,1
1,CE,CE,CE,CE,CE,CE,CE,CE,C1,C1,C
1,C1,C1,C1,CE,CE,C1,C1,C1,C1,C1,C
1,CE,CE
1870 DATA C1,C1,C1,C1,C1,C1,CE,C
E,F1,F1,F1,F1,F1,F1,71,71,F1,F1,F
1,F1,F1,F1,71,71,61,61,61,61,61,4
1,41,41
1880 DATA 61,61,61,61,61,61,61,6
1,A1,A1,A1,A1,A1,A1,A1,A1,A1,A
1,A1,A1,A1,A1,A1,F1,F1,F1,F1,F1,F
1,F1,F1
1890 DATA F1,F1,F1,F1,F1,F1,F1,F
1,B1,B1,B1,B1,B1,51,51,51,B1,B1,B
1,B1,B1,B7,B7,57,51,51,51,51,51,5
1,51,51
1900 DATA 51,5b,5b,51,51,51,d1,5
1,61,61,61,61,61,61,61,61
1910 ' *********************
1920 DATA 00,03,07,01,00,00,00,0
0,00,03,03,00,02,03,00,00,7c,fc,f
6,f2,78,78,78,70,00,a8,f8,3c,7c,f
e,00,00
1930 DATA 00,00,00,00,00,05,0d,0
7,07,00,00,01,01,00,00,00,00,00,0
0,00,00,80,80,80,80,00,00,80,80,0
0,00,00
1940 DATA 00,00,00,0e,0f,00,00,0
0,00,00,38,2e,20,00,00,00,00,00,0
0,00,80,00,00,00,78,50,00,40,00,0
0,70,f0
1950 DATA 3f,3f,6f,4f,1e,1e,1e,0
e,00,15,1f,3c,3e,7f,00,00,00,c0,e
0,80,00,00,00,00,00,c0,c0,00,40,c
0,00,00
1960 DATA 00,00,00,00,00,01,01,0
1,01,00,00,01,01,00,00,00,00,00,0
0,00,00,A0,b0,e0,e0,00,00,80,80,0
0,00,00
1970 DATA 00,00,00,00,01,00,00,0
0,1e,0a,00,02,00,00,0e,0f,00,00,0
0,70,f0,00,00,00,00,00,1c,74,04,0
0,00,00
1980 DATA 03,0f,1f,3c,28,08,08,0
0,00,07,07,07,0f,1f,00,00,c0,e0,f
0,18,08,08,08,00,00,f8,e0,60,e8,f
8,00,00
1990 DATA 00,00,00,00,00,02,06,0
3,03,00,08,08,00,00,00,00,00,00,0
0,00,00,a0,b0,e0,e0,00,04,04,00,0
0,00,00
2000 DATA 00,00,00,03,07,04,00,0
c,04,00,00,00,00,00,07,0f,00,00,0
0,e0,f0,10,00,18,10,00,18,9a,10,0
0,70,78
2010 DATA 38,44,42,42,21,1b,07,0
f,08,0e,1e,2f,2f,06,03,00,0e,11,2
1,21,42,ec,f0,f8,08,b8,bc,fa,fa,3
0,e0,00
2020 DATA 00,00,00,00,00,01,03,0
3,03,01,00,38,74,62,60,00,00,00,0
0,00,00,c0,e0,e0,e0,c0,80,8e,17,2
3,03,00
```



CLASE DE ECUACIONES

Clase: Educativo

COMO EJEMPLO :
VAMOS A DESPEJAR B

Para ello utilizamos el

B = C

A

Lo estas haciendo bien,
Pulsa M para retorno a

El profesor de Química y Merceología, Roberto Emilio Gaguine, de la Escuela Nacional de Educación Técnica Nº 1 de San José de la Quintana, nos ha hecho llegar un interesante programa creado por él.

Su experiencia en programas aplicados a la educación puede ser interesante para los lectores de nuestra revista.

Roberto envió este programa para compartirlo con todos los lectores y para pedir nuestra opinión sobre el mismo.

Nos pareció muy completo, claro y que cumple el objetivo de reforzar los conocimientos sobre el tema. Sabemos que no es sencillo dar una clase sobre "Ecuaciones", pero ésta es bastante completa y didáctica.

Daremos una sugerencia que facilitara el manejo de los programas.

Es conveniente colocar "mensajes aclaratorios" en la pantalla sobre el siguiente paso que debe cumplir el operador para continuar con la ejecución del programa.

Por ejemplo, en este programa, para pasar de explicación se debe presionar cualquier tecla.

Sería conveniente colocar un cartel para sacar la duda de con qué se debe continuar.

Otro detalle que se podría mejorar está la opción de "Despeje de la incógnita". Al despejar correctamente la computadora debería aclararlo y pasar automáticamente al menú principal.

Pero éstos son sólo detalles que no hacen desmejorar el programa.

Destacamos la claridad en la explicación de los conceptos.

Realmente es un programa muy bien pensado y diseñado.

Aclaramos a los lectores que tiene siete opciones donde se explican los conceptos básicos sobre las "ecuaciones" además de graficarlas.

Por último agradecemos a Roberto por su programa, manteniendo abierta la comunicación con él. Nos interesan sus dudas, sugerencias, comentarios y programas (como también los de los demás lectores, porque ellos pueden ayudar a todos los usuarios de las MSX).

VARIABLES IMPORTANTES:

A$: almacena la opción elegida del menú principal
B: valor de b para la opción "Resolución de ecuaciones"
C: valor de c para la misma opción
D: valor de d para la opción anterior
X: resultado de x
X1: primera raíz de la opción "Resolución de una ecuación"
X2: segunda raíz para la misma opción
A: coeficiente del término para la última opción
F: coeficiente del término lineal
C: término independiente de la ecuación para la misma opción

ESTRUCTURA DEL PROGRAMA:

50-110: pantalla inicial
120-280: menú principal
290-350: procesos para la primera opción
360-560: segunda opción
570-840: tercera opción
850-2020: cuarta opción
2030-3020: quinta opción
3030-3460: sexta opción
3470-3830: séptima opción

```
10 REM **************************
****
20 REM   PROFESOR ROBERTO E GAGU
INE
30 REM **************************
****
40 REM ******** ECUACIONES *****
****
50 SCREEN 3:COLOR 4,0,0:CLS
60 OPEN "GRP:" AS#1:KEY OFF
70 PLAY"L4CEDFDECREFGAGFER"
80 PRESET(30,20):PRINT#1,"ECUA"
90 PRESET(105,80):PRINT#1,"CIO"
100 PRESET(145,140):PRINT#1,"NES
"
110 IF INKEY$="" THEN 110
120 CLS:SCREEN 0:COLOR 1,15,15
130 LOCATE 4,0:PRINT"ECUACIONES"
140 LOCATE 4,1:PRINT"[redacted]"
150 LOCATE 4,3:PRINT"MENU":BEEP
160 LOCATE 4,5:PRINT"1 CONCEPTO"
:BEEP
170 LOCATE 4,7:PRINT"2 GRADO DE
UNA ECUACION":BEEP
180 LOCATE 4,9:PRINT"3 CLASIFICA
CION DE LAS ECUACIONES":BEEP
190 LOCATE 4,11:PRINT"4 DESPEJE
DE LA INCOGNITA":BEEP
```




```
200 LOCATE 4,13:PRINT"5 RESOLUCI
ON DE ECUACIONES DE":BEEP
210 LOCATE 4,14:PRINT"  1er.GRAD
O CON UNA INCOGNITA":BEEP
220 LOCATE 4,16:PRINT"6 RESOLUCI
ON DE ECUACIONES DE":BEEP
230 LOCATE 4,17:PRINT"  2do.GRAD
O CON UNA INCOGNITA":BEEP
240 LOCATE 4,19:PRINT"7 RECONSTR
UCCION DE UNA ECUACION        CUAD
RATICA.":BEEP
250 A$=INKEY$
260 IF A$="" THEN 250
270 IF A$<"1" OR A$>"7" THEN 250
280 ON VAL(A$) GOTO 290,360,570,
850,2030,3030,3470
290 CLS:SCREEN 0
300 LOCATE 5,5:PRINT"CONCEPTO DE
 ECUACION":BEEP
310 LOCATE 5,6:PRINT"██████████
█████"
320 IF INKEY$="" THEN 320
330 LOCATE 0,10:PRINT"LLAMAREMOS
 ECUACION,A LAS IGUALDADES":BEEP:
LOCATE 0,12:PRINT"MATEMATICAS QUE
 SOLO SE VERIFICAN PA-":BEEP:LOCA
TE 0,14:PRINT"RA DETERMINADOS VAL
ORES ATRIBUIDOS A ":BEEP:LOCATE 0
,16:PRINT"SUS INCOGNITAS,DESIGNAD
AS,AQUI,CON X.":BEEP
340 IF INKEY$="" THEN 340
350 GOTO 120
360 CLS:COLOR 1,15,15
370 LOCATE 4,2:PRINT"GRADO DE UN
A ECUACION":BEEP
380 LOCATE 4,3:PRINT"███████████
█"
390 LOCATE 0,6:PRINT"EL GRADO DE
 UNA ECUACION ESTA DADO   POR EL
MAYOR EXPONENTE DE SUS INCOGNITAS
.":BEEP
400 IF INKEY$="" THEN 400
410 LOCATE 4,10:PRINT"EJEMPLOS:"
:BEEP:LOCATE 4,11:PRINT"├├├├├├├├"
420 LOCATE 4,11:PRINT"├├├├├├├"
430 LOCATE 4,15:PRINT"8X - 5 + 3
X = 16"
440 IF INKEY$="" THEN 440
450 LOCATE 0,13:PRINT"ECUACION D
E 1er.GRADO CON 1 INCOGNITA":BEEP
460 IF INKEY$="" THEN 460
470 LOCATE 4,19:PRINT"5X² - 12 +
 2X = X² - 3"
480 IF INKEY$="" THEN 480
490 LOCATE 0,17:PRINT"ECUACION D
E 2do.GRADO CON 1 INCOGNITA":BEEP
500 IF INKEY$="" THEN 500
510 LOCATE 4,24:PRINT"X + 3 = 2X
"
520 LOCATE 5,21:PRINT"3"
530 IF INKEY$="" THEN 530
540 LOCATE 0,20:PRINT"ECUACION D
E 3er.GRADO CON 1 INCOGNITA":BEEP
550 IF INKEY$="" THEN 550
560 GOTO 120
570 CLS:COLOR 1,15,15
580 LOCATE 2,2:PRINT"CLASIFICACI
ON DE LAS ECUACIONES":BEEP
590 LOCATE 2,3:PRINT"███████████
██████████████████"
600 IF INKEY$="" THEN 600
610 LOCATE 0,6:PRINT"LAS ECUACIO
NES SE CLASIFICAN EN ENTE RAS,FRA
CCIONARIAS E IRRACIONALES.":BEEP
620 IF INKEY$="" THEN 620
630 LOCATE 0,12:PRINT"LA ECUACIO
N ES ENTERA CUANDO LAS IN- CONGIT
AS ESTAN AFECTADAS POR LAS OPE RA
CIONES DE SUMA,RESTA Y MULTIPLICA
- CION.":BEEP
640 IF INKEY$="" THEN 640
650 LOCATE 4,18:PRINT"EJEMPLO:"
660 LOCATE 4,19:PRINT"├├├├├├├"
670 LOCATE 4,21:PRINT"4 _ 5X =
1 + 48"
680 LOCATE 13,22:PRINT"├├├":LOCA
TE 14,23:PRINT"3"
690 IF INKEY$="" THEN 690
700 CLS:LOCATE 0,4:PRINT"UNA ECU
ACION ES FRACCIONARIA CUANDO  POR
 LO MENOS UNA DE LAS INCOGNITAS
 FIGURA COMO DIVISOR.":BEEP
710 IF INKEY$="" THEN 710
720 LOCATE 4,11:PRINT"EJEMPLO:"
730 LOCATE 4,12:PRINT"├├├├├├├"
740 LOCATE 4,14:PRINT" 3  + 8X _
 5 = 34"
750 LOCATE 4,15:PRINT"├├├"
760 LOCATE 4,16:PRINT" X "
770 IF INKEY$="" THEN 770
780 CLS:LOCATE 0,4:PRINT"UNA ECU
ACION ES IRRACIONAL CUANDO PORLO
MENOS UNA INCOGNITA FIGURA BAJO
SIGNO RADICAL.":BEEP
790 IF INKEY$="" THEN 790
800 LOCATE 4,10:PRINT"POR EJEMPL
O:"
810 LOCATE 4,11:PRINT"├├├├├├├├├├
├"
820 LOCATE 4,14:PRINT"1 - √X = 8
 - 16"
830 IF INKEY$="" THEN 830
840 GOTO 120
850 CLS:COLOR 1,15,15
860 LOCATE 0,2:PRINT"REGLA EN QU
E SE BASA LA RESOLUCION DEECUACIO
NES DE 1er.GRADO CON UNA INCOGNIT
A.":BEEP
870 IF INKEY$="" THEN 870
880 LOCATE 4,6:PRINT"REGLA 1"
890 LOCATE 4,7:PRINT"├├├├├├├"
900 IF INKEY$="" THEN 900
910 LOCATE 0,8:PRINT"SI SE SUMA
O SE RESTA UN MISMO NUMEROA AMBOS
 MIEMBROS DE UNA ECUACION,SE  OBT
IENE OTRA ECUACION EQUIVALENTE A
 LA DADA.":BEEP
920 IF INKEY$="" THEN 920
930 LOCATE 4,13:PRINT"SEA LA ECU
ACION:"
940 LOCATE 4,14:PRINT"├├├├├├├├├├
├├├├├"
950 LOCATE 4,16:PRINT"8X = 2X +
7   CUYA RAIZ ES X=7/6"
960 IF INKEY$="" THEN 960
970 LOCATE 0,18:PRINT"SI A AMBOS
 MIEMBROS LE SUMAMOS UN MISMO NUM
ERO,3 POR EJEMPLO,OBTENEMOS:":BEE
P
980 LOCATE 4,21:PRINT"8X + 3 = 2
X + 7 + 3"
990 IF INKEY$="" THEN 990
1000 LOCATE 0,23:PRINT"QUE ES EQ
UIVALENTE A LA PRIMERA PUES SU RA
IZ ES X=7/6 ."
1010 IF INKEY$="" THEN 1010
1020 CLS:LOCATE 4,0:PRINT"TOMEMO
S EL EJEMPLO ANTERIOR"
1030 LOCATE 4,2:PRINT"8X = 2X +
7 ":BEEP
1040 LOCATE 0,4:PRINT"SI A ESTA
ECUACION ANTERIOR,LE RESTA-MOS,PO
R EJEMPLO 7,A AMBOS MIEMBROS"
1050 IF INKEY$="" THEN 1050
1060 LOCATE 0,8:PRINT"8X - 7 = 2
X + 7 - 7":BEEP
1070 IF INKEY$="" THEN 1070
1080 LOCATE 0,10:PRINT"REDUCIEND
O +7 Y -7 EN EL SEGUNDO MIEMBRO,N
OS QUEDA:"
1090 LOCATE 0,13:PRINT"8X - 7 =
2X":BEEP
1100 IF INKEY$="" THEN 1100
1110 LOCATE 4,15:PRINT"*********
***"
1120 LOCATE 4,16:PRINT" CONCLUSI
ON"
1130 LOCATE 4,17:PRINT"*********
***"
1140 LOCATE 0,19:PRINT"TODO NUME
RO QUE FIGURA SUMANDO EN UN MIEMB
RO DE UNA ECUACION,PUEDE PASAR  A
L OTRO MIEMBRO RESTANDO Y RECIPRO
CA-MENTE,SI ESTA RESTANDO PUEDE P
ASAR SUMANDO AL OTRO MIEMBRO.":BE
EP
1150 IF INKEY$="" THEN 1150
1160 CLS:LOCATE 4,0:PRINT"REGLA
2"
1170 LOCATE 4,1:PRINT"├├├├├├├"
1180 LOCATE 0,3:PRINT"SI SE MULT
IPLICAN O SE DIVIDEN AMBOS MIEMBR
OS DE UNA ECUACION POR UN MISMONU
MERO O FACTOR,QUE NO SEA NULO,SE
OBTIENE OTRA ECUACION EQUIVALENTE
 A LA DADA."
1190 IF INKEY$="" THEN 1190
1200 LOCATE 4,10:PRINT"POR EJEMP
LO:"
1210 LOCATE 5,12:PRINT"5X + 2 =
7X"
1220 IF INKEY$="" THEN 1220
1230 LOCATE 0,14:PRINT"ES UNA EC
UACION DE RAIZ X=1
1240 IF INKEY$="" THEN 1240
1250 LOCATE 0,16:PRINT"SI A AMBO
S MIEMBROS LOS MULTIPLICAMOSPOR U
N MISMO NUMERO,POR EJEMPLO 3,OB-T
ENDREMOS OTRA ECUACION EQUIVALENT
E,OSEA DE RAIZ X=1":BEEP
1260 IF INKEY$="" THEN 1260
1270 LOCATE 0,21:PRINT"(5X + 2).
3 = 7X.3":BEEP
1280 IF INKEY$="" THEN 1280
1290 CLS:LOCATE 0,1:PRINT"TOMEMO
S POR EJEMPLO:"
1300 IF INKEY$="" THEN 1300
1310 LOCATE 4,3:PRINT"7X - 2 = 5
"
1320 LOCATE 4,4:PRINT"├├├├├├"
1330 LOCATE 4,5:PRINT"    6"  :BE
EP
1340 IF INKEY$="" THEN 1340
1350 LOCATE 0,7:PRINT"MULTIPLICA
NDO A AMBOS MIEMBROS POR 6 SE OBT
IENE:"
1360 IF INKEY$="" THEN 1360
1370 LOCATE 4,10:PRINT"7X - 2 .6
 = 5.6"
1380 LOCATE 4,11:PRINT"├├├├├├"
1390 LOCATE 4,12:PRINT"    6    ";
BEEP
1400 IF INKEY$="" THEN 1400
1410 LOCATE 0,14:PRINT"SIMPLIFIC
ANDO OBTENEMOS:"
1420 LOCATE 4,15:PRINT"7X - 2 =
5.6":BEEP
1430 IF INKEY$="" THEN 1430
1440 LOCATE 4,17:PRINT"*********
***"
1450 LOCATE 4,18:PRINT" CONCLUSI
ON"
1460 LOCATE 4,19:PRINT"*********
***"
1470 LOCATE 0,21:PRINT"SI UN NUM
ERO ESTA MULTIPLICANDO A TO-DO UN
 MIEMBRO EN UNA ECUACION,PUEDE  P
ASAR AL OTRO MIEMBRO COMO DIVISOR
 Y RECIPROCAMENTE, SI ESTA DIVIDI
ENDO A TODO UN MIEMBRO PUEDE PASA
R MULTIPLI-CANDO A TODO EL OTRO"
1480 IF INKEY$="" THEN 1480
1490 CLS: LOCATE 4,0:PRINT"CONSE
CUENCIA"
1500 LOCATE 4,1:PRINT"███████████
██"
1510 LOCATE 0,3:PRINT"SE PUEDE C
AMBIAR EL SIGNO A TODOS LOSMIEMBR
OS DE UNA ECUACION.ESTO SE VERIFI
CA MULTIPLICANDO A AMBOS MIEMBROS
  POR (-1)":BEEP
1520 LOCATE 11,10:PRINT"-2X + 4
= -10"
1530 LOCATE 4,13:PRINT"(-2X + 4)
.(-1) = (-10).(-1)
1540 LOCATE 12,16:PRINT"2X - 4 =
 10":BEEP
1550 IF INKEY$="" THEN 1550
1560 COLOR 1,10,10:CLS:SCREEN 2,
3
1570 PRESET(15,0):PRINT#1,"COMO
EJEMPLO :"
1580 PRESET(15,10):PRINT#1,"VAMO
S A DESPEJAR B":BEEP
1590 PRESET(5,30):PRINT#1,"Para
ello utilizamos el cursor."
1600 PRESET(15,180):PRINT#1,"Pul
sa M para retorno al MENU"
1610 PRESET(20,100):PRINT#1,"███
"
1620 PRESET(120,100):PRINT#1,"="
1630 PRESET(130,100):PRINT#1," █
█"
1640 PRESET(60,88):PRINT#1,"X"
1650 IF INKEY$="" THEN 1650
1660 JJ=JJ+1
1670 IF JJ=1 THEN 1680 ELSE 1730
```

```
1680 A$=A$+CHR$(&H18)
1690 B$=B$+CHR$(&H3C)
1700 C$=C$+CHR$(&H66)
1710 D$=C$:E$=E$+CHR$(&H7E)
1720 F$=E$:G$=C$:H$=C$
1730 SPRITE$(0)=A$+B$+C$+D$+E$+F$+G$+H$:IF JJ=1 THEN 1740 ELSE 1800
1740 AQ$=AQ$+CHR$(&HFE)
1750 AW$=AW$+CHR$(&HFF)
1760 AE$=AE$+CHR$(&HE7)
1770 AR$=AW$:AZ$=AW$
1780 AT$=AE$
1790 AY$=AW$:AU$=AQ$
1800 SPRITE$(1)=AQ$+AW$+AE$+AR$+AZ$+AT$+AY$+AU$:IF JJ=1 THEN 1810 ELSE 1840
1810 X=27:Y=77
1820 AA$=AA$+CHR$(&HFE)
1830 BB$=AA$:CC$=AA$:DD$=DD$+CHR$(&HE0):EE$=DD$:FF$=AA$:GG$=AA$:HH$=AA$
1840 SPRITE$(2)=AA$+BB$+CC$+DD$+EE$+FF$+GG$+HH$
1850 C=STICK(0)
1860 PUT SPRITE 0,(X,Y),15
1870 PUT SPRITE 1,(82,79),6
1880 PUT SPRITE 2,(160,79),4
1890 IF C=1 THEN Y=Y-1
1900 IF C=2 THEN X=X+1:Y=Y-1
1910 IF C=3 THEN X=X+1:IF X=100 THEN COLOR 10,1,10:PSET(60,88):PRINT#1,"X":PSET(20,100):PRINT#1,"[redacted]":COLOR 1,10,10
1920 IF C=4 THEN X=X+1:Y=Y+1
1930 IF C=5 THEN Y=Y+1
1940 IF C=6 THEN X=X-1:Y=Y+1
1950 IF C=7 THEN X=X-1
1960 IF C=8 THEN X=X-1:Y=Y-1
1970 IF X>138 AND Y>100 THEN 1980 ELSE 1990
1980 PRESET(5,170):PRINT#1,"Lo estas haciendo bien,continúa"
1990 M$=INKEY$
2000 IF M$<>"M" THEN 1850
2010 ON VAL(M$) GOTO 2020
2020 GOTO 120
2030 CLS:SCREEN 0
2040 PRINT"PROGRAMA PARA RESOLVER UNA ECUACION "
2050 PRINT"DE PRIMER GRADO CON UNA INCOGNITA"
2060 IF INKEY$="" THEN 2060
2070 LOCATE 5,4
2080 PRINT"POR EJEMPLO LA ECUACION:"
2090 LOCATE 5,6
2100 PRINT"(x + b).c = d"
2110 IF INKEY$="" THEN 2110
2120 LOCATE 9,8
2130 PRINT"x + b = d "
2140 LOCATE 17,9
2150 PRINT"├├"
2160 LOCATE 17,10
2170 PRINT"c"
2180 IF INKEY$="" THEN 2180
2190 LOCATE 13,12
2200 PRINT"x = d    b"
2210 LOCATE 17,13
2220 PRINT"├├ ├"
2230 LOCATE 17,14
2240 PRINT"c"
2250 IF INKEY$="" THEN 2250
2260 LOCATE 13,16
2270 PRINT"x = "
2280 IF INKEY$="" THEN 2280
2290 LOCATE 5,19
2300 INPUT"INGRESE b:";B
2310 LOCATE 5,21:INPUT"INGRESE c:";C
2320 LOCATE 5,23:INPUT"INGRESE d:";D
2330 X=(D/C)-B
2340 LOCATE 16,15:PRINT X
2350 IF INKEY$="" THEN 2350
2360 COLOR 1,3,3
2370 CLS:SCREEN 2:COLOR 1,3,3
2380 PRESET(5,10):PRINT#1,"GRAFICO DE LA FUNCION :"
2390 PRESET(5,20):PRINT#1,"[redacted]"
2400 PRESET(100,28):PRINT#1,"(X + 3). 2 = Y":BEEP
2410 LINE(150,50)-(150,180),4:LINE(90,110)-(210,110),4:BEEP
2420 IF INKEY$="" THEN 2420
2430 FOR Q=60 TO 105 STEP 5
2440 PSET(150,Q),15
2450 NEXT Q
2460 FOR W=140 TO 160 STEP 5
2470 PSET(W,110),15
2480 NEXT W
2490 PRESET(147,42):PRINT#1,"Y":BEEP
2500 PRESET(213,107):PRINT#1,"X":BEEP:PRESET(76,108):PRINT#1,"-X":BEEP:PRESET(146,170):PRINT#1,"-Y":BEEP
2510 IF INKEY$="" THEN 2510
2520 LINE(40,50)-(40,120):LINE(20,60)-(60,60)
2530 IF INKEY$="" THEN 2530
2540 PRESET(27,53):PRINT#1,"X":BEEP:PSET(47,53):PRINT#1,"Y":BEEP
2550 IF INKEY$="" THEN 2550
2560 PRESET(27,62):PRINT#1,"1   8":BEEP
2570 IF INKEY$="" THEN 2570
2580 PRESET(155,70),4
2590 IF INKEY$="" THEN 2590
2600 PRESET(27,74):PRINT#1,"2   10":BEEP
2610 IF INKEY$="" THEN 2610
2620 PRESET(160,60),4
2630 IF INKEY$="" THEN 2630
2640 PRESET(27,86):PRINT#1,"0   6":BEEP
2650 IF INKEY$="" THEN 2650
2660 PRESET(150,80),6
2670 IF INKEY$="" THEN 2670
2680 PRESET(20,98):PRINT#1,"-1   4":BEEP
2690 IF INKEY$="" THEN 2690
2700 PSET(145,90),4
2710 IF INKEY$="" THEN 2710
2720 PRESET(20,110):PRINT#1,"-2   2":BEEP
2730 IF INKEY$="" THEN 2730
2740 PSET(140,100),4
2750 IF INKEY$="" THEN 2750
2760 LINE(170,40)-(120,140),4
2770 LINE(170,40)-(120,140),4
2780 IF INKEY$="" THEN 2780
2790 PRESET(5,175):PRINT#1,"LA RECTA CORTA AL EJE X EN(3,0)":BEEP
2800 PRESET(5,185):PRINT#1,"E INTERSEPTA AL EJE Y EN (0,6)":BEEP
2810 IF INKEY$="" THEN 2810
2820 CLS:SCREEN 0:COLOR 1,3,3
2830 LOCATE 4,10:PRINT"aX + b = Y"
2840 LOCATE 0,3:PRINT"VOY A GRAFICAR UNA FUNCION DE PRIMER GRADO CON UNA INCOGNITA,POR FAVOR"
2850 LOCATE 0,19:PRINT"Para graficar utilizaré la siguiente"
2860 LOCATE 0,21:PRINT"ESCALA :1 división cada 5 pixels"
2870 LOCATE 4,6:INPUT"INGRESE A";A
2880 LOCATE 4,7:INPUT"INGRESE B";B
2890 IF INKEY$="" THEN 2890
2900 CLS:SCREEN 2
2910 LINE (60,100)-(180,100),4:PRESET(45,90):PRINT#1,"-X":PRESET(190,90):PRINT#1,"X":BEEP:LINE(120,20)-(120,190),4:PRESET(117,10):PRINT#1,"Y":PRESET(105,180):PRINT#1,"-Y":BEEP
2920 FOR E=60 TO 180 STEP 5:FOR R=20 TO 190 STEP 5
2930 PSET(E,100),15:PSET(120,R),15
2940 NEXT R :NEXT E
2950 PRESET(10,0):PRINT#1,"La escala es 1 en 5"
2960 IF INKEY$="" THEN 2960
2970 FOR X=-50 TO 100
2980 Y=-A*X-B
2990 PSET STEP(120,100),1:PSET(X,Y),3
3000 NEXT X
3010 IF INKEY$="" THEN 3010
3020 GOTO 120
3030 CLS:SCREEN 0
3040 PRINT"PROGRAMA PARA RESOLVER UNA ECUACION"
3050 PRINT"DE SEGUNDO GRADO ."
3060 LOCATE 6,5:PRINT"ax²+ bx + c = 0"
3070 LOCATE 0,10
3080 INPUT"COEFICIENTE DEL TERMINO CUADRATICO a:";A
3090 LOCATE 0,13
3100 INPUT"COEFICIENTE DEL TERMINO LINEAL b:";B
3110 LOCATE 0,15
3120 INPUT"TERMINO INDEPENDIENTE c:";C
3130 O=(B^2)-(4*A*C)
3140 IF O<0 THEN 3180 ELSE 3150
3150 H=O^(1/2):V=B*(-1)+H:T=-B-H:X1=V/(2*A):X2=T/(2*A)
3160 PRINT:PRINT:PRINT"X1=";X1
3170 PRINT:PRINT"X2=";X2:GOTO 3200
3180 PRINT:PRINT:PRINT"NO PUEDO RESOLVERLA EN EL CAMPO REAL"
3190 PRINT:PRINT"LO SIENTO"
3200 IF INKEY$="" THEN 3200
3210 CLS:SCREEN 0
3220 PRINT"VOY A GRAFICAR UNA FUNCION CUADRATI- CA,POR FAVOR":BEEP
3230 LOCATE 0,14:PRINT"Le aconsejo que el valor de A esté"
3240 LOCATE 0,15:PRINT"comprendido entre -1 y 1,y el de B"
3250 LOCATE 0,16:PRINT"que no supere +-10,tratando que el va"
3260 LOCATE 0,17:PRINT"lor de C no sea muy grande"
3270 LOCATE 0,20:PRINT"ESTO SE DEBE A QUE LA PANTALLA ES"
3280 LOCATE 0,21:PRINT"MUY REDUCIDA":BEEP
3290 LOCATE 4,5:INPUT"INGRESE A";A
3300 LOCATE 4,7:INPUT"INGRESE B";B
3310 LOCATE 4,9:INPUT"INGRESE C";C
3320 CLS:SCREEN 2
3330 LINE(60,100)-(180,100),4:PRESET(185,90):PRINT#1,"X":PRESET(45,90):PRINT#1,"-X":BEEP
3340 LINE(120,20)-(120,190),4:PRESET(117,10):PRINT#1,"Y":PRESET(105,180):PRINT#1,"-Y":BEEP
3350 IF INKEY$="" THEN 3350
3360 FOR W=60 TO 180 STEP 5:FOR Q=20 TO 190 STEP 5
3370 PSET(W,100),15:PSET(120,Q),15
3380 NEXT Q:NEXT W
3390 IF INKEY$=""THEN 3390
3400 FOR X=-50 TO 100
3410 PSET STEP(120,100),1
3420 Y=-A*X^2-B*X-C
3430 PSET(X,Y),3
3440 NEXT X
3450 IF INKEY$="" THEN 3450
3460 GOTO 120
3470 COLOR 1,15,15
3480 CLS:COLOR 1,15,15:SCREEN 0
3490 LOCATE 4,0:PRINT"RECONSTRUCCION DE UNA ECUACION "
3500 LOCATE 4,1:PRINT"├├├├├├├├├├├├├├├├├├├├├├├├├├├├├├"
3510 LOCATE 6,3:PRINT"CUADRATICA"
3520 LOCATE 6,4:PRINT"├├├├├├├├├├"
3530 IF INKEY$="" THEN 3530
3540 LOCATE 0,10:PRINT"LOS VALORES DE LAS RAICES DEBERA IN-"
3550 LOCATE 0,12:PRINT"TRODUCIRLOS COMO NUMEROS RACIONALES":BEEP
3560 IF INKEY$="" THEN 3560
3570 CLS
3580 LOCATE 4,0:PRINT"RECONSTRUCCION DE UNA ECUACION"
3590 LOCATE 4,1:PRINT"├├├├├├├├├├├├├├├├├├├├├├├├├├├├├├"
3600 LOCATE 6,3:PRINT"CUADRATICA"
3610 LOCATE 6,4:PRINT"├├├├├├├├├├"
3620 IF INKEY$="" THEN 3620
3630 LOCATE 0,5:PRINT"INTRODUZCA LAS RAICES DE LA ECUACION"
3640 LOCATE 4,7:INPUT"VALOR DE X1=";M
3650 LOCATE 17,8:PRINT"├├├"
3660 LOCATE 16,9:INPUT"";N
3670 LOCATE 4,11:INPUT"VALOR DE X2=";V
3680 LOCATE 17,12:PRINT"├├├"
3690 LOCATE 16,13:INPUT"";J
3700 LOCATE 4,15:PRINT"LA ECUACION RECONSTRUIDA ES:"
3710 A=N*J
3720 B=(A/N)*M+(A/J)*V
3730 C=M*V
3740 F=-B
3750 LOCATE 4,16:PRINT"AX² + BX + C = 0"
3760 LOCATE 1,18:PRINT"SIENDO A ="
3770 LOCATE 11,18:PRINT A
3780 LOCATE 8,19:PRINT"B ="
3790 LOCATE 11,19:PRINT F
3800 LOCATE 8,20:PRINT"C ="
3810 LOCATE 11,20:PRINT C:BEEP
3820 IF INKEY$="" THEN 3820
3830 GOTO 120
```

LAS RIENDAS DEL ASSEMBLER (2ª parte)

A partir de este momento comenzamos a sumergirnos dentro del Z-80.
Veremos su funcionamiento y cómo controlar sus actividades.

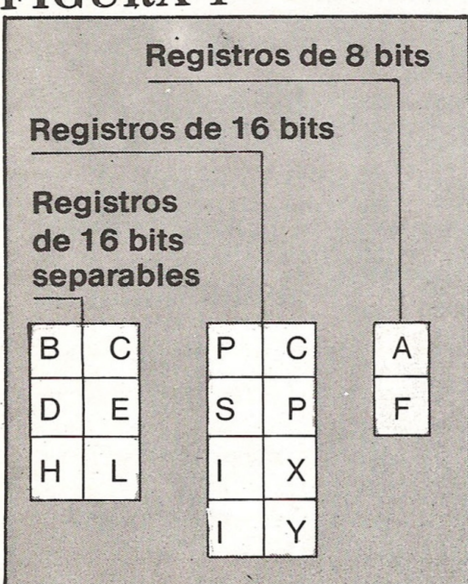


En el número anterior vimos cómo es que un microprocesador genérico de 8 bits, trabaja numéricamente.
Obviamente, todavía nos queda bastante por hablar al respecto, pero es suficiente como para comenzar.

LOS REGISTROS

Estos son lugares dentro del mismísimo Z-80, en los cuales podremos almacenar temporalmente alguna información
Este espacio es muy similar (de hecho podemos pensarlo igual) al de una memoria RAM de 26 bytes.
Al ser tan pequeña la capacidad del mismo, estos bytes o pares de bytes pueden llevar nombres diferentes. Obviamente sería desesperante llamar a cada uno de los 65535 bytes de la RAM con un nombre distinto.

FIGURA 1



Cada uno de los bytes o pares asociados de bytes, dentro de esos 26 que posee el Z-80 en su interior, son los famosos registros.

Así, para acceder a uno de estos registros, no usaremos su número sino su nombre.
Además, algunos de éstos tiene funciones determinadas. Es decir que para realizar determinadas tareas el Z-80 reconoce solamente a algunos de ellos.
En la figura 1 observamos los registros que generalmente utilizaremos. Como vemos, están separados en grupos: los de 8 bits, los de 16 bits (dos bytes), y los de 16 bits separables.
Los registros llamados A y F son dos registros de 8 bits (ibyte) cada uno, pudiendo almacenar 256 números distintos.
Los registros PC, SP, IX, e IY son registros de 16 bits (2 bytes) a los que accederemos con sólo invocarlos, como veremos más adelante, y que pueden almacenar 65536 números diferentes.
Por último los registros BC, DE, y HL son un poco especiales.

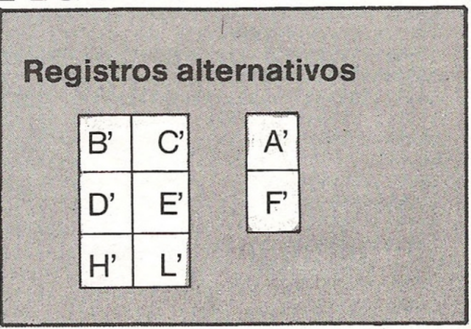
Por ejemplo, el par BC es un registro de 16 bits que al igual que el PC, puede almacenar 65536 números distintos. Pero (recordemos que son dos bytes BC), también este par puede funcionar como dos registros de 8 bits cada uno, el B y el C.

Lo mismo ocurre para DE y HL. Esto puede no ser demasiado claro, pero podemos pensarlo como que algunas instrucciones (que veremos) utilizan sólo uno, o sea la mitad de estos registros, por ejemplo el E. Así E será un registro, o un byte bien deteminado dentro del Z-80.

PROFUNDICEMOS

Vamos a ver cómo es que un número es almacenado dentro de uno de estos registros de 16 bits.
Supongamos que de alguna forma podemos llenar el par BC con el número 512.
Si con una lupa informática pudiéramos ver sólo el registro B de ese par, veríamos el número 2, y examinando el registro C veríamos un 0 (cero).
¿Que ocurrió?. Si efectuamos la siguiente multiplicación tal vez se nos aclare un poco el panorama:
B * 256 + C = 512
B en el caso de pensar al par BC como de 16 bits, es el llamado byte **Alto** y C es el llamado byte **Bajo.**
Si sólo queremos almacenar un número pequeño, no mayor que 255, podremos hacerlo, por ejemplo en un registro con el B. Pero si nuestro número a almacenar supera dicha cantidad, deberemos hacerlo en uno como el BC. En sí, siguen siendo dos registros de 8 bits pero con un significado distinto, uno vale 256 veces más que el otro. Esto nos permitirá almacenar hasta la cifra que nombramos anteriormente (65535).
Este proceso no puede efectuarse con los registros no separables (PC, SP, IX, e IY). Supongamos de igual forma que almacenamos en el registro H el número 4 y en el L el número 1. Así si observamos el registro H veremos efectivamente que contiene el valor 4 y el L el valor 1, pero si agrandamos la lupa, y observamos el par completo HL como un registro de 16 bits, veremos que contiene, al número 1025. Esto es así, pues:
H*256+L=
=4*256+1=1025
El almacenamiento en los registros A y F no ofrece mayores dificultades de comprensión, basta con saber lo que vimos en el número anterior, pues cada uno representa a un byte común y silvestre.
Por otro lado existen los llamados registros alternativos, que vemos en la figura 2. Estos son registros que usa el Z-80 independientemente para efectuar

FIGURA 2



las operaciones necesarias, para cumplir con la intrucción que le mandamos hacer. En síntesis, no se pueden tocar si no se es un experto en el tema. Por otra parte no tendremos necesidad de hacerlo. De hecho, un mal manejo de estos registros pueden causar efectos insospechados.
Pero presentemos formalmente a cada uno de estos registros.

EL ACUMULADOR

El emperador dentro de la MSX, el Z-80, posee variadas instrucciones y variados **grupos** de instrucciones.
La gran mayoría de las operaciones matemáticas, lógicas o comparativas, entre otras, usan el acumulador como registro de referencia.
El acumulador es nada más y nada menos que el registro A.
Este humilde y privilegiado registro tiene 8 bits como capacidad de almacenamiento.

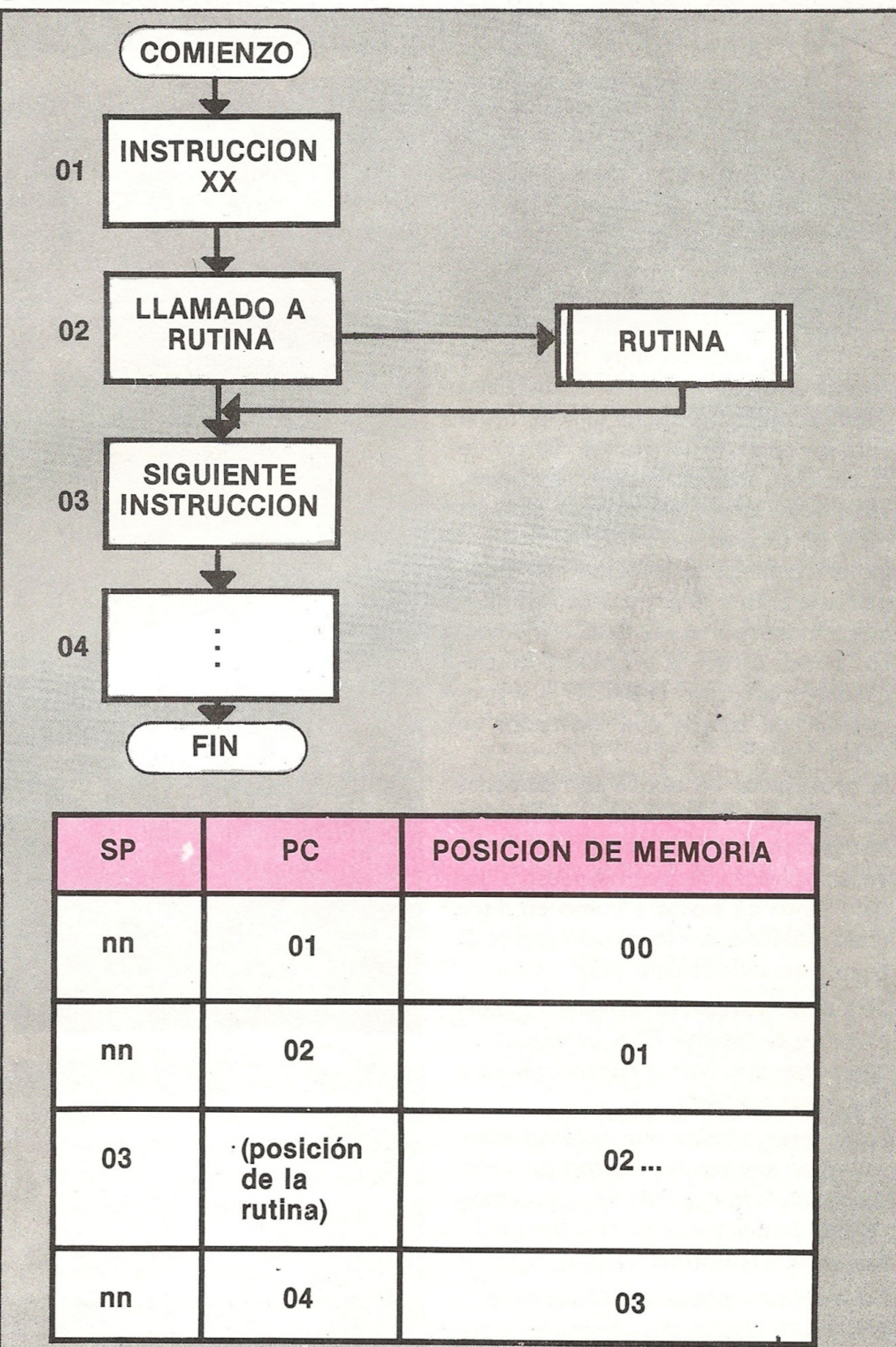
El PC

Este es un registro de 16 bits (dos bytes) no separable con una característica muy particular. Sin él, la capacidad de ejecutar un programa de cualquier clase sería nula.
Es el encargado de "almacenar" el valor de la posición de memoria, en el cual se encuentra la sentencia que se está ejecutando de un programa Assembler. ¿Fácil, no?.
Pensemos que el micro debe recordar, en todo momento, cuál es la siguiente instrucción a ejecutar. De lo contrario se perdería y los efectos de ello ya sabemos cuáles son.
Su nombre (PC) hace referencia a las palabras inglesas: Program Counter.
Así, la forma en que afectaremos el contenido de este registro será principalmente la que puede causar la longitud de nuestro programa.
Como vimos, sólo puede contener como máximo el valor 65535, lo que nos indica a las claras que una instrucción no puede alojarse en una posición de memoria mayor que este valor. Por eso, nuestro programa Assembler nunca podrá tener una longitud mayor a los 65535 bytes (64 Kbytes). Este es el motivo por el cual las MSX, generalmente y sin artilugios de paginación o solapamiento de RAM, no pueden "direccionar" más de 64 Kbytes por vez.

SP

Stack Pointer es la raíz de su nombre, y al igual que el anterior, su capacidad de almacenamiento es de 16 bits (2 bytes) y no es separable en dos registros de 8 bits.
Su función tiene que ver con su nombre, Stack (pila) y Pointer (puntero).
Este registro es el encargado de contener el valor de la posición de memoria en la cual se halla el comienzo de una imaginaria "pila" de datos. Se puede decir que el SP "apunta" al comienzo de la "pila". En su momento veremos cómo se accede a ella.
Por otra parte, esta pila tiene muchos usos, pero en forma general podemos decir que su función es almacenar valores temporales, que deben recordarse para su posterior uso.
La pila no es más que una zona perdida en la RAM de nuestra MSX, y su uso ha de ser cuidadoso, pues nuestro programa (si es que la utiliza) deberá compartirla con el mismísimo Z-80.
Por ejemplo, cada vez que un programa Assembler llama a una rutina, el valor de la posición de memoria en la cual se aloja la siguiente instrucción a ejecutar luego de esa rutina, debe ser almacenado en algún lugar para recordarlo a su debido tiempo. Ese lugar es la "pila".
En la figura 3, vemos gráficamente la ejecución de un programa imaginario, que nos muestra lo que acabamos de decir.
En el próximo número continuaremos viendo los restantes registros y la nomenclatura utilizada.

FIGURA 3



SP	PC	POSICION DE MEMORIA
nn	01	00
nn	02	01
03	(posición de la rutina)	02 ...
nn	04	03

AVANZA LA NORMA MSX

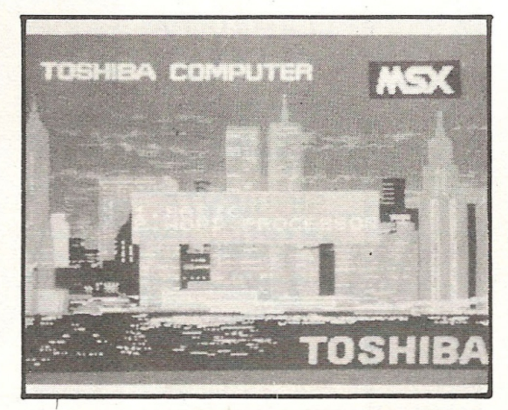
Con la aparición de esta máquina en el mercado, la compatibilidad del standard MSX se extiende en nuestro país. La posibilidad de utilizar la memoria como RAM-disk, y dos conectores para cartuchos son ventajosas novedades en la computadora de Toshiba. Pero la HX-20 se hace sobresaliente al presentar también el procesador de textos incluído.

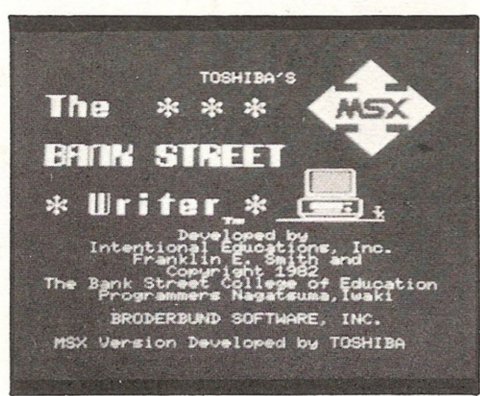
Su teclado, tipo máquina de escribir, es totalmente negro al igual que la carcaza. El set de teclas especiales se diferencia por su color gris azulado, salvando la tecla de STOP, que está acabada en rojo fuerte.
Sin entrar en sus especificaciones técnicas, nos damos cuenta de que esta computadora es diferente. Sólo encendiéndola, percibimos que se ha creado con un criterio estético muy moderno y acorde a lo que este tipo de máquinas puede ofrecer al usuario. Una hermosa vista en color de una costa repleta de rascacielos, muy parecida a Manhattan, visto desde Brooklyn, irrumpe ante los ojos del usuario.
Inmediatamente aparece un menú que marca la primer diferencia con las MSX que transitan por el país hasta el momento. Se nos ofrece la posibilidad de elegir entre dirigirnos al BASIC o al procesador de textos que incorpora en ROM.
El procesador de textos es realmente una herramienta potente, con muchas instrucciones de fácil manejo, que incluye la posibilidad de archivar nuestro trabajo tanto en cassette como en disco (cosa que no todos los procesadores de textos conocidos para MSX poseen).

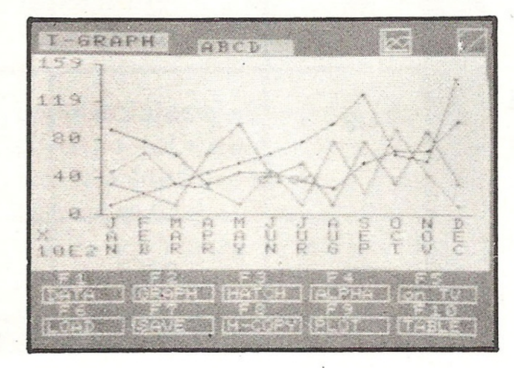
Otra de las novedades (siguiendo las normas que el sistema MSX propone), es que posee dos slots o conectores para cartuchos standard.

La norma no indica que necesariamente tengan que ser dos los slots para cartucho, pero lo que indica es que si existe más de una salida de este tipo, debe ajustarse a la norma.

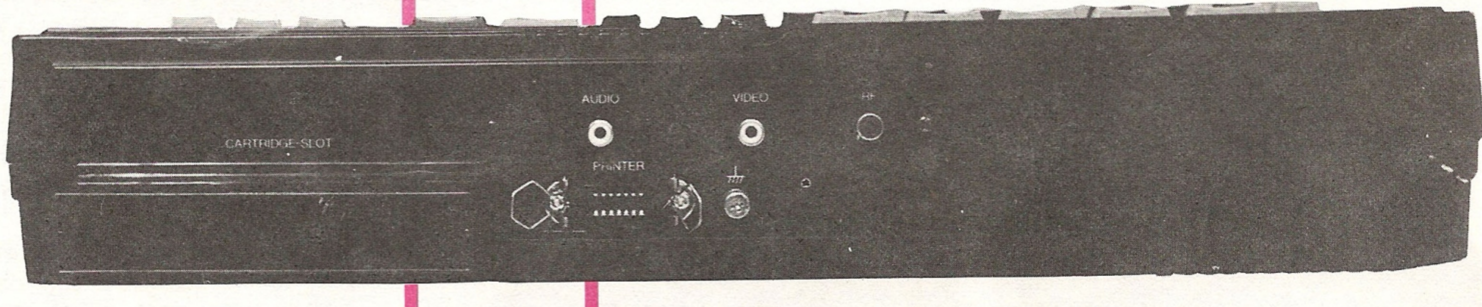
Por supuesto posee sus teclas de cursor, en un diseño elegante y cómodo, sin resaltar demasiado las del resto pues su color es el mismo que el de las teclas especiales.
Entre sus accesorios, posee una pequeña disquetera de moderno formato que utiliza discos de 3'' y media, que si bien son más caros que los de 5'' y 1/4, son mucho más fiables. Basta con ver los últimos diseños de disqueteras para las más

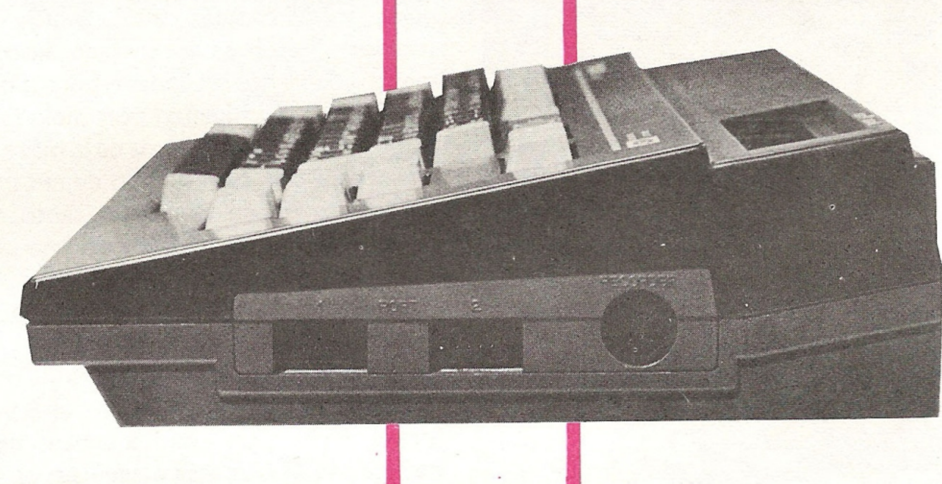






afamadas computadoras para notar que esto es así. Por ejemplo, la línea ST de ATARI utiliza estos discos al igual que la PC Convertible de IBM y muchas otras. Estos disquetes vienen encapsulados en un sobre de plástico rígido, y la ranura por la que el cabezal del drive penetra, está protegida con una cubierta metálica retráctil que el mismo drive se encarga de correr para que pueda tomar contacto su cabezal con el disco.
Por otro lado, estos discos (aunque parezca mentira) son capaces de soportar más información que los anteriores.
Aunque la **HX-F101** (así se llama la disquetera) sólo soporta, ateniéndose a las normas, 360 Kbytes en cada disco, éstos son capaces de soportar algo más de 1 Mbyte sin formatear.
Continuando con los medios de almacenamiento, y aunque puede utilizar cualquier grabador de cassettes para pasar archivos, esta firma saca a la venta un **Data Recorder KT-P22** para su uso.
Además, puede ser usado con cualquier computadora.
Pero el tema de los medios de almacenamiento no termina aquí.
Veamos. Este tipo de máquinas (las MSX) sólo pueden trabajar con 64 Kbytes de memoria sea RAM o ROM debido a la limitación que produce el Z-80, que es su microprocesador central.
Así, por ejemplo, podremos tener un bloque de 32 Kbytes de ROM que contenga el MSX-BASIC y el procesador de textos y 32 Kbytes de RAM. De esta forma nos quedan 32 Kbytes de RAM libres, pues esta máquina trae 64 Kbytes de RAM, y hasta aquí no hay nada novedoso. Pero resulta que mientras trabajamos cómodamente con el MSX-BASIC standard, podremos utilizar esos 32 Kbytes que la mayoría de las computadoras emplea para cargar un sistema operativo como el MSX-DOS u otro programa, para almacenar nuestros propios programas.
¿Cómo? Es fácil. Resulta, y esto sí es novedad, que esta computadora posee otros 32 Kbytes de ROM en los cuales se aloja lo que ellos llaman un Basic ampliado.
Entre otras cosas, nos da acceso a una serie de instrucciones que permiten grabar nuestros archivos en la RAM, que antes no podíamos utilizar, como si fuera un drive de discos.





Los que no podamos adquirir un drive junto con la máquina, sí podremos almacenar nuestros programas en esa RAM, mientras trabajamos con otros, o simplemente con nuevas versiones de nuestro programa. Así, al finalizar el día, se pueden bajar los archivos allí alojados a cassette o a disquete.

Como ya lo estarán suponiendo, la velocidad de transferencia, o el tiempo que tarda en pasar un programa a esta RAM-DISK, es prácticamente imperceptible, pues ocurre a la velocidad de procesamiento del Z-80, olvidándonos de la velocidad de los motores del drive de disquetes.

Por otro lado, las instrucciones con que contamos para el tratamiento de los archivos, son prácticamente todas e iguales a las de cualquier sistema de discos de MSX standard (MSX-DOS).

Por supuesto que no encontraremos las instrucciones que vuelcan un sector del disco a RAM, pues esto no es un disquete.

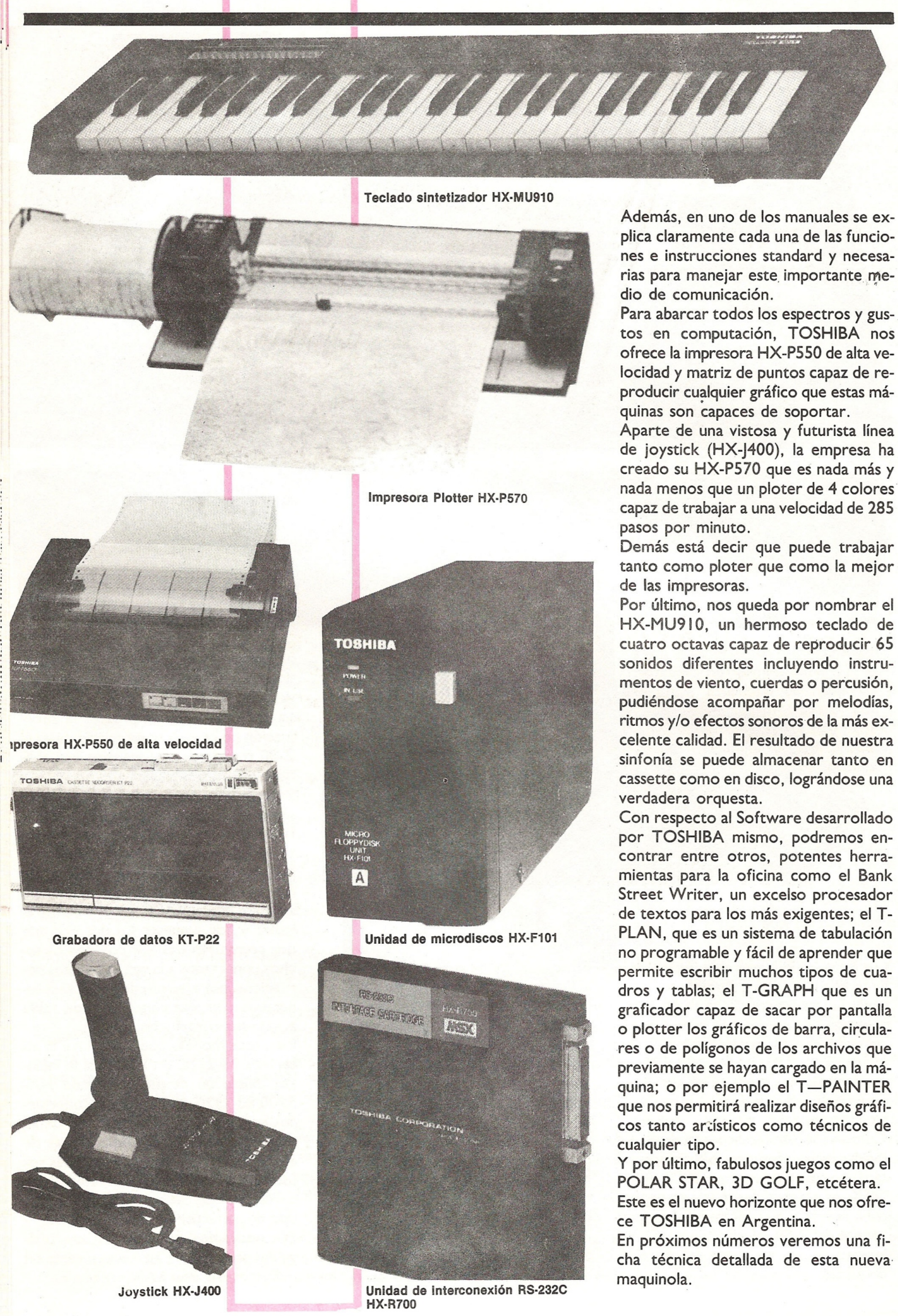
Continuando con la memoria, podemos decir que posee sus reglamentarios 16 Kbytes de RAM para video, sumando un total de 80 Kbytes de RAM y 64 Kbytes de ROM.

Por el lado de la información, no le falta nada, pues viene acompañada por tres manuales que explican todo su funcionamiento, desde una introducción al Basic para principiantes hasta el manejo de su RAM-DISK.

La velocidad con que procesa la información no es nada despreciable, pues el Z-80 trabaja en esta máquina casi al límite de sus posibilidades :3,58 Mhz.

Por supuesto posee además la salida de video compuesto, RF para TV, audio, y la salida paralelo standard para conectar una impresora cualquiera de este tipo o lo que se nos ocurra. Además, los dos conectores de joystick, también standard (tipo ATARI), la salida para grabador y por supuesto los dos slot que nos permitirán trabajar con dos cartuchos conectados a la vez, o con dos periféricos, por ejemplo dos drives sin tener que comprar ningún tipo de cable adaptador o cartucho.

Para este modelo que no incluye la interface RS-232C (el HX-22 sí la incluye) existe un cartucho llamado HX-R700 RS-232C, que posibilita conectarnos con cualquier periférico que tenga este tipo de interface (impresoras de cualquier marca, modem —del más sofisticado al más simple—, etcétera). Recordemos que cualquier PC posee este tipo de conector de comunicaciones, y que podremos conectarnos con cualquier computadora en forma directa sin utilizar redes tipo LAN.



Teclado sintetizador HX-MU910

Impresora Plotter HX-P570

presora HX-P550 de alta velocidad

Grabadora de datos KT-P22

Unidad de microdiscos HX-F101

Joystick HX-J400

Unidad de interconexión RS-232C HX-R700

Además, en uno de los manuales se explica claramente cada una de las funciones e instrucciones standard y necesarias para manejar este importante medio de comunicación.

Para abarcar todos los espectros y gustos en computación, TOSHIBA nos ofrece la impresora HX-P550 de alta velocidad y matriz de puntos capaz de reproducir cualquier gráfico que estas máquinas son capaces de soportar.

Aparte de una vistosa y futurista línea de joystick (HX-J400), la empresa ha creado su HX-P570 que es nada más y nada menos que un ploter de 4 colores capaz de trabajar a una velocidad de 285 pasos por minuto.

Demás está decir que puede trabajar tanto como ploter que como la mejor de las impresoras.

Por último, nos queda por nombrar el HX-MU910, un hermoso teclado de cuatro octavas capaz de reproducir 65 sonidos diferentes incluyendo instrumentos de viento, cuerdas o percusión, pudiéndose acompañar por melodías, ritmos y/o efectos sonoros de la más excelente calidad. El resultado de nuestra sinfonía se puede almacenar tanto en cassette como en disco, lográndose una verdadera orquesta.

Con respecto al Software desarrollado por TOSHIBA mismo, podremos encontrar entre otros, potentes herramientas para la oficina como el Bank Street Writer, un excelso procesador de textos para los más exigentes; el T-PLAN, que es un sistema de tabulación no programable y fácil de aprender que permite escribir muchos tipos de cuadros y tablas; el T-GRAPH que es un graficador capaz de sacar por pantalla o plotter los gráficos de barra, circulares o de polígonos de los archivos que previamente se hayan cargado en la máquina; o por ejemplo el T—PAINTER que nos permitirá realizar diseños gráficos tanto artísticos como técnicos de cualquier tipo.

Y por último, fabulosos juegos como el POLAR STAR, 3D GOLF, etcétera.

Este es el nuevo horizonte que nos ofrece TOSHIBA en Argentina.

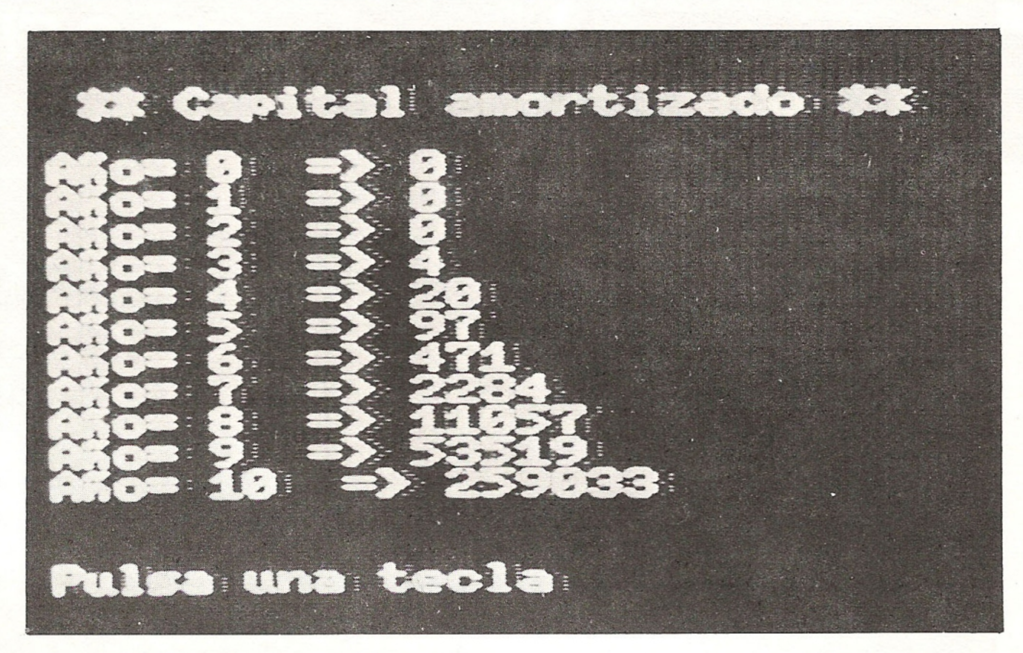
En próximos números veremos una ficha técnica detallada de esta nueva maquinola.



PRESTAMOS Y AMORTIZACIONES

CLASE COMERCIAL

Para ampliar el uso de las MSX, en este número proponemos un programa sobre capitales y préstamos.
Se puede realizar en cuatro sistemas: francés, uniforme, alemán y americano.
Explicaremos brevemente las diferencias entre estos sistemas.
Los dos primeros son bastante similares pues la amortización del préstamo se basa en cuotas iguales en donde se comienza con un interés que irá luego decreciendo. Esto se debe a que el interés se calcula sobre el saldo.
En cambio para el sistema alemán la cuota de interés es decreciente y se amortiza sobre el mismo capital.
En el sistema americano, el interés es igual para todos los años.
El programa informa sobre cuotas anuales sin interés, las cuotas de amortización del capital prestado, el capital amortizado y otros detalles más.
Se trata de un completo software, muy útil para los que deban realizar operaciones comerciales sobre préstamos.





Variables importantes:

CC: capital
N: años que dura el préstamo
TI: tipo de interés (porcentaje)
TM: interés en el mercado
A: matriz con las anualidades
CA: matriz con cuotas de amortización
I: matriz con cuotas de interés
CV: matriz con capital vivo
FA: matriz con fondo de amortización
FI: matriz con fondo de interés
DP: matriz con deuda pendiente

Estructura del programa:

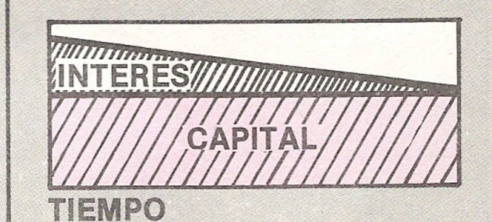
10-230: pantalla inicial y menú de opciones
240-370: cálculo para el sistema francés
380-550: cálculos para el sistema alemán
560-690: cálculos para el uniforme
700-870: cálculos para el americano
880-1330: muestra todos los resultados para cada sistema
1340-1400: opciones finales

Gráfico comparativo entre el capital y el interés

SISTEMA UNIFORME Y FRANCES

INTERES
CAPITAL
TIEMPO

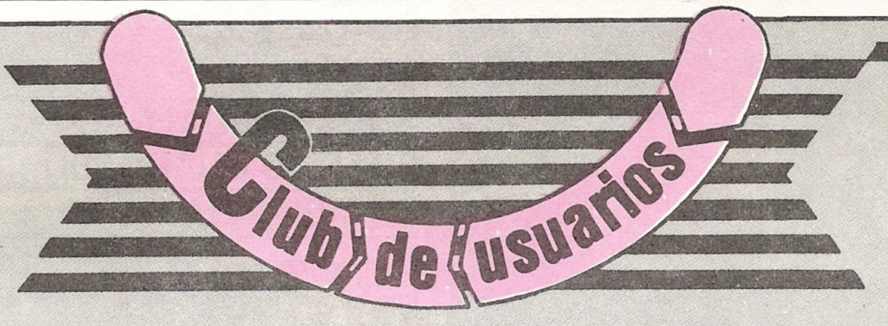
SISTEMA ALEMAN



TIEMPO

```
10 CLS:KEY OFF:WIDTH 40:COLOR 11
,1
20 FOR K=1 TO 10
30 LOCATE 7,22:PRINT"Amortizació
n de préstamos"
40 FOR H=1 TO 100:NEXT H
50 CLS:FOR H=1 TO 100:NEXT H
60 NEXT K
70 LOCATE 7,22:PRINT"Amortizació
n de préstamos"
80 FOR K=1 TO 500:NEXT K
90 FOR K=1 TO 18:PRINT:NEXT K
100 LOCATE 0,7
110 PRINT"    1-Sistema francés."
120 PRINT"    2-Sistema alemán."
130 PRINT"    3-Sistema uniforme"
140 PRINT"    4-Sistema americano
"
150 PRINT:PRINT:PRINT"     ELIGE
UNA OPCION  (1-4)"
160 Z$=INPUT$(1)
170 Z=VAL(Z$):IF Z>4 OR Z<1 THEN
 150
180 ON Z GOSUB 240,380,560,700
190 CLS:PRINT:PRINT
200 INPUT" Quieres hacer otro cu
adro?";N$
210 N$=LEFT$(N$,1)
220 IF N$="s" OR N$="S" THEN RUN
```

```
230 END
240 CLS:PRINT
250 INPUT"Ingresa el capital pre
stado.                  cap:";CC
260 INPUT"Ingresa el número de a
ños que duró          año:";N
270 INPUT"Ingresa el tipo de int
erés.                Tipo:";TI
280 IF TI>1 THEN TI=TI/100
290 CV(0)=CC
300 FOR K=1 TO N
310 A(K)=CC*((1+TI)^N*TI)/((1+TI
)^N-1)
320 I(K)=CV(K-1)*TI
330 CA(K)=A(K)-I(K)
340 E(K)=E(K-1)+CA(K)
350 CV(K)=CC-E(K)
360 NEXT K:SIST=0
370 GOSUB 880:RETURN
380 CLS:PRINT
390 INPUT"Ingresa el capital pre
stado.                  cap:";CC
400 INPUT"Ingresa el número de a
ños que duró          año:";N
410 INPUT"Ingresa el tipo de int
erés.                Tipo:";TI
420 IF TI>1 THEN TI=TI/100
430 CV(0)=CC:I(0)=CC*TI
440 FOR K=1 TO N
450 A(K)=(CC*TI)/(1-(1-TI)^N)
460 NEXT K:CA(N)=A(N)
470 FOR K=N-1 TO 1 STEP -1
480 CA(K)=CA(K+1)*(1-TI)
490 NEXT K
500 FOR K=1 TO N
510 E(K)=E(K-1)+CA(K)
520 CV(K)=CC-E(K)
530 I(K)=CV(K)*TI
540 NEXT K:SIST=0
550 GOSUB 880:RETURN
560 CLS:PRINT
570 INPUT"Ingresa el capital pre
stado.                  cap:";CC
580 INPUT"Ingresa el número de a
ños que duró          año:";N
590 INPUT"Ingresa el tipo de int
erés.                Tipo:";TI
600 IF TI>1 THEN TI=TI/100
610 CV(0)=CC
620 FOR K=1 TO N
630 CA(K)=CC/N
640 E(K)=E(K-1)+CA(K)
650 CV(K)=CC-E(K)
660 I(K)=CV(K)*TI
670 A(K)=I(K)+CA(K)
680 NEXT K:SIST=0
690 GOSUB 880:RETURN
700 CLS:PRINT
710 INPUT"Ingresa el capital pre
stado.                  cap:";CC
720 INPUT"Ingresa el número de a
ños que duró          año:";N
730 INPUT"Ingresa el tipo de int
erés.                Tipo:";TI
740 INPUT"Ingresa el tipo de int
erés del mercado.    Tipo:";TM
750 IF TI>1 THEN TI=TI/100
760 IF TM>1 THEN TM=TM/100
770 DP(0)=CC
780 FOR K=1 TO N
790 I(K)=CC*TI
800 CA(K)=CC*TM/((1+TM)^N-1)
810 A(K)=I(K)+CA(K)
820 FA(K)=CA(K)+FA(K-1)*(1+TM)
830 FI(K)=FA(K-1)*TM
840 DP(K)=CC-FA(K)
850 NEXT K
860 SIST=4
870 GOSUB 880:RETURN
880 CLS:PRINT
890 PRINT" ** Anualidades **":PR
INT
900 FOR K=0 TO N
910 PRINT"Año=";K;" =>";INT(A(K)
)
920 NEXT K
930 GOSUB 1370
940 PRINT" ** Cuotas de amortiza
ción **":PRINT
950 FOR K=0 TO N
960 PRINT"Año=";K;" =>";INT(CA(K
))
970 NEXT K
980 GOSUB 1370
990 PRINT" ** Cuotas de interés
**":PRINT
1000 FOR K=0 TO N
1010 PRINT"Año=";K;" =>";INT(I(K
))
1020 NEXT K
1030 GOSUB 1370
1040 IF SIST=4 THEN 1170
1050 PRINT" ** Capital amortizad
o **":PRINT
1060 FOR K=0 TO N
1070 PRINT"Año=";K;" =>";INT(E(K
))
1080 NEXT K
1090 GOSUB 1370
1100 PRINT" ** Capital vivo **":
PRINT
1110 FOR K=0 TO N
1120 PRINT"Año=";K;" =>";INT(CV(
K))
1130 NEXT K
1140 GOSUB 1370
1150 GOSUB 1340
1160 RETURN
1170 PRINT" ** Fondo de amortiza
ción **":PRINT
1180 FOR K=0 TO N
1190 PRINT"Año=";K;" =>";INT(FA(
K))
1200 NEXT K
1210 GOSUB 1370
1220 PRINT" ** Interés del fondo
 **":PRINT
1230 FOR K=0 TO N
1240 PRINT"Año=";K;" =>";INT(FI(
K))
1250 NEXT K
1260 GOSUB 1370
1270 PRINT" ** Deuda pendiente *
*":PRINT
1280 FOR K=0 TO N
1290 PRINT"Año=";K;" =>";INT(DP(
K))
1300 NEXT K
1310 GOSUB 1370
1320 GOSUB 1340
1330 RETURN
1340 PRINT:PRINT
1350 INPUT "Deseas ver otra vez
los resultados";N$:N$=LEFT$(N$,1)
1360 IF N$="s" OR N$="S" THEN 88
0
1370 PRINT:PRINT
1380 PRINT"Pulsa una tecla"
1390 IF INKEY$="" THEN 1390
1400 CLS:PRINT:RETURN
```



Si bien este espacio está reservado para informaciones acerca del Club MSX, es de nuestro interés interiorizarlos sobre el funcionamiento del CEDI (Centro para el Desarrollo de la Inteligencia). En él se desarrollan diversas actividades referentes a la educación utilizando como herramienta fundamental al computador. Cursos, talleres y seminarios son algunas de las actividades con las que cuenta esta Institución, los cuales se llevan a cabo con el respaldo de profesionales idóneos en distintas áreas (licenciados en sistemas, analistas de sistemas, pedagogos, psicólogos, etc.), lo cual asegura una gran responsabilidad, tanto sea en la preparación de los programas de estudio, dictado de cursos, elección de técnicas pedagógicas adecuadas y todo aquello concerniente al proceso de enseñanza-aprendizaje.

Pero todas nuestras actividades no están centradas en un segmento de la población en especial, sino por el contrario estamos abiertos a todo aquel que desee, de alguna manera, introducirse en el mundo de la Informática. Para ello contamos con planes apropiados que puedan satisfacer las distintas expectativas planteadas por docentes, profesionales, niños, jóvenes y adultos.

A su vez el CEDI, a través de todo su personal trabaja en forma constante para ofrecer un mejor y más completo servicio. De esta manera es que el Club los invita a acercarse a este Establecimiento e informarse sobre los nuevos planes de cursos de verano. Los mismos están diagramados de tal forma que, por su duración y frecuencia no le exigirán alterar el normal desarrollo de sus actividades.

NUEVA SEDE EN CORDOBA 650 — CAP. FED.

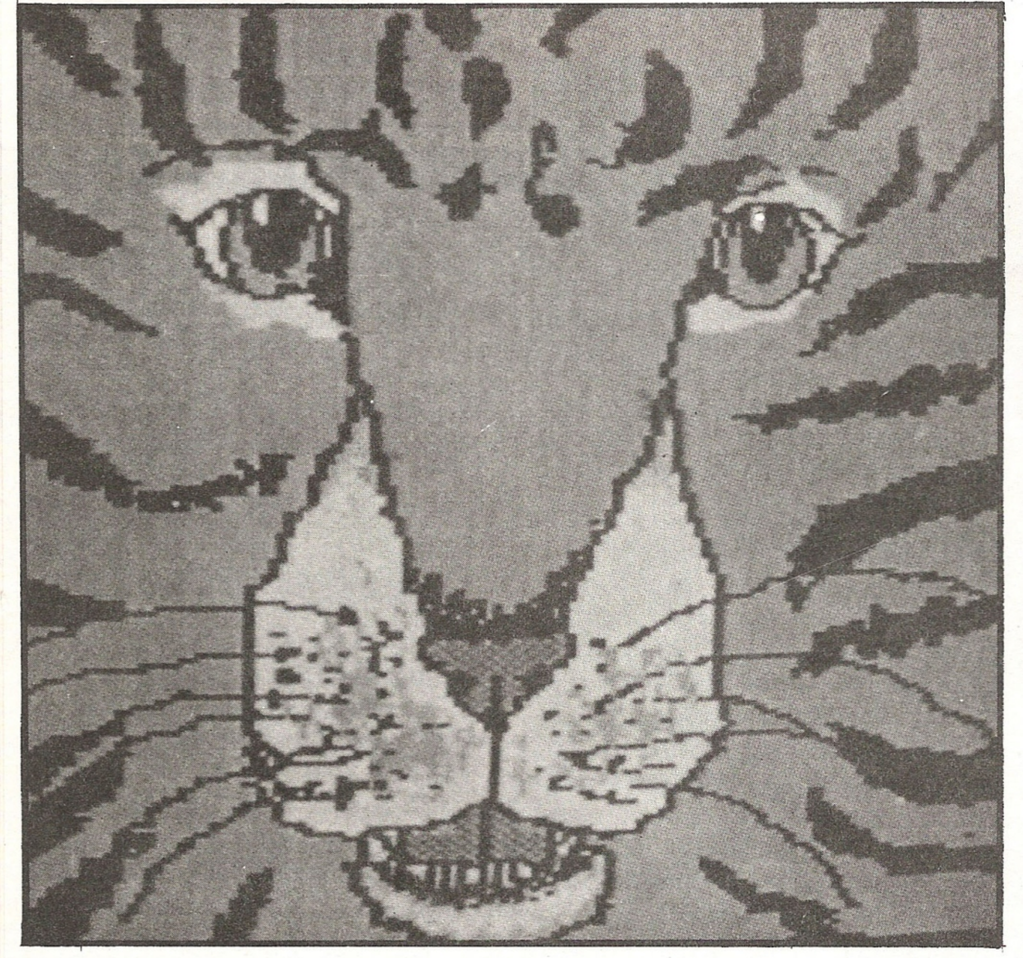
Teniendo en cuenta las consultas de socios y usuarios en general, sobre la posibilidad de abrir una nueva sede ubicada en el microcentro de la ciudad, es que nos es grato informales sobre la inauguración de un nuevo Centro adherido a nuestro Club. Este Centro, que cuenta con instalaciones de primer nivel y una cantidad importante de equipos y soft, es una prueba de que el Club de Usuarios MSX tiene la más firme ambición de brindarle cada día un servicio más eficiente.
Este Centro, además de funcionar como Club de Usuarios MSX, proveerá los servicios del Centro para el Desarrollo de la Inteligencia (CEDI).

NUEVO SOFT DISPONIBLE:

- **Lenguaje de autor Pilot**
- **Lenguaje Prolog**
- **Nuevos utilitarios**
- **Lenguaje Cobol-80**

GRAFICADOR ALTA RESOLUCION

Como vimos en los números anteriores, hemos programado pedacitos del programa: cómo movernos por la pantalla, pintar y de qué forma podemos guardar nuestro gráfico.
Pero con esto no hemos terminado de completar la estructura que habíamos planteado desde el comienzo. Restan detalles como los que veremos a continuación.



CORTAR LA CONTINUIDAD DEL DIBUJO

Como vamos guardando los puntos estratégicos de nuestro diseño, uno a continuación del otro, luego, para reconstruir el gráfico, solamente uniremos cada punto con sus vecinos. Es decir, el primer punto se unirá con el segundo, al mismo tiempo que éste se unirá con el tercero y así sucesivamente.

Pero supongamos que tenemos diseñado un dibujo como el de la figura 11. En esta figura mostramos lo que podría ser nuestra pantalla y los puntos que deben guardar nuestras matrices PX y PY. Pero para desunir a los puntos PX(3), PY(3) con PX(4), PY(4), deberemos agregar un indicador. Por ejemplo: presionar otra tecla de función provocando el almacenamiento en cada matriz de un número especial, como el "–1" (Ver figura 12).
Por lo tanto solo se deberá colocar, para esta parte del programa, las siguientes sentencias:

aumentar el número
de casillero .... A=A+1
si no llenamos la
matriz .... IF A<601
entonces .... THEN
colocamos el número
especial en
las dos...PX(A)= –1 : PY(A)= –1
matrices
volvemos al programa ... RETURN
principal

Esta es otra parte del programa y la uniremos luego con las otras porciones

BORRAR CUALQUIER GRAFICO GUARDADO EN MEMORIA

Cuando queremos borrar los puntos que fuimos entrando en la matriz al presionar alguna tecla de función, significa en realidad borrar los valores que almacenaban en las cajas o bien colocar a cada una el valor cero (0).
Como sabemos que el último elemento (el último casillero ocupado) es el que tiene el número de orden A, haremos un loop desde 1 hasta A colocando el valor cero en el casillero de PX y al mismo tiempo en el de PY:

hacemos un loop
desde 1 hasta A .. FOR F=1 TO A
guardaremos un cero
en cada casillero...PX(F)=0: PY(F)=0
sucesivamente
aumentar la variable
del loop .... NEXT
volver al programa ...... RETURN
principal

REPRODUCIR NUESTRO DISEÑO

Vimos que esto se realiza uniendo cada

punto con sus vecinos. Entonces al teclear una tecla de función:

pintaremos de color
rojo oscuro
el primer ... PSET (PX(1), PY(1)),6
punto del gráfico
con un loop desde 2
hasta A .... FOR F=2 TO A
si está guardado
el número especial .. IF PX(F)= – 1
entonces .... THEN
salto un casillero
(se cortó el dibujo)
y pinto el punto que
representa..PSET(PX(F+1),PY(F+1)),6
las coordenadas de
PX y PY
incrementar el loop .... NEXT
volver al programa ...... RETURN

Todas estas sentencias se colocan una a continuación de la otra. Pero cuando se da el caso en que PX(F) sea diferente de "– 1", el controlador del programa saltará a la siguiente línea. Esta será:

unir el punto que se
está leyendo . LINE–(PX(F),PY(F),6
con el punto anterior
incrementar el loop .... NEXT
volver al programa ...... RETURN
principal

Con esto hemos terminado con los bloques del programa y ahora faltarían algunos detalles que ayudarán a unirlos.

LEVANTAR EL LAPIZ

A medida que movamos la flecha por la pantalla, iremos dejando puntos pintados con el color rojo oscuro. Pero si deseamos movernos sin modificar con esto el gráfico que estemos creando, esto se lograría al colocar un marcador o bandera.

Por ejemplo, al presionar la barra de espacio, encenderemos la bandera o marcador poniéndole el valor – 1. Si volvemos a presionarla, la bandera tendrá el valor 1. Cuando se encuentre el valor del marcador en – 1, le indicará a la computadora NO pintar nuestros pasos, y hacerlo si el valor es 1.

Transformémoslo en sentencia:
cambiar el signo de
la bandera .... NF=NF*(–1)
(porque se presionó
la barra espaciadora)

Con esta expresión matemática, NF alternará su valor entre "1" ó "– 1" a medida que utilicemos la barra
volver al programa ...... RETURN

MOVER EL SPRITE

Para ubicar el sprite (flecha en nuestro caso) según nuestros movientos con los cursores (vimos esta parte en el número anterior) se emplea la sentencia: PUT SPRITE número de sprite, (columna, fila), color del sprite. Para nuestro programa, el sprite definido lleva el número cero (0), columna será el valor de la variable X y fila el de la variable Y. Pintaremos de color blanco al sprite, cuyo código es 15.

Finalmente la sentencia será: PUT SPRITE 0, (X,Y), 15.

PINTAR CADA PUNTO

Directamente lo pasaremos a sentencia:
si la bandera es 1 .... IF NF=1
entonces .... THEN
pinto el punto donde se
encuentra actualmente.. .PSET (X,Y)
el sprite

En cambio si la bandera es diferente de "1" se continúa con el programa.

DETERMINAR LAS TECLAS PARA CADA TAREA

Hemos detallado cada bloque y dijimos que se ejecutarán al presionar las diferentes teclas de función.

Figura 11

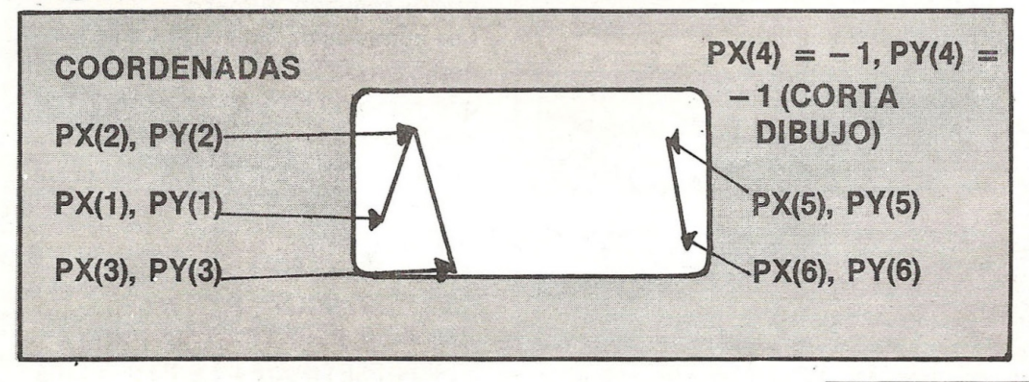


Figura 12

COORDENADAS
PX(2), PY(2)
PX(1), PY(1)
PX(3), PY(3)
PX(4) = – 1, PY(4) = – 1 (CORTA DIBUJO)
PX(5), PY(5)
PX(6), PY(6)

Por ejemplo destinemos:

F1 .. borrará punto por punto
F2 .. borrará regiones que encierren los cuadriláteros
F3 .. guarda los puntos estratégicos del diseño
F4 .. corta la continuidad del dibujo
F5 .. borra cualquier gráfico guardado en memoria
F6 .. reproduce nuestro diseño

Mientras que al teclear la barra de espacio levantaremos el lápiz.

Estas asignaciones se declaran de la siguiente forma:

ON STRIG GOSUB número de línea.

Esto significa que si se presiona la barra de espacio, el controlador del programa saltará al número de línea declarado en esa instrucción y volverá cuando encuentre una sentencia RETURN. En cambio ON KEY GOSUB num. línea 1, num. línea 2, num línea 3,... hará que al presionarse la primer tecla de función (F1) salte a "num. línea 1" (que en realidad es un número y no letras, en nuestro programa es la línea 6200), realice todas las operaciones hasta una sentencia RETURN y vuelva. Si es presionada la tecla F2, saltará a num. línea 2, línea 6210 para nuestro programa, y así se sigue.

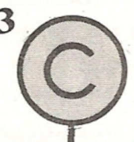
Solamente resta preparar el sistema de la siguiente forma:

habilitar las teclas de .... FOR F=1 TO 6 : KEY (F) ON: NEXT
función (sólo 6 funciones)
habilitar la barra de .. STRIG(0) ON
espacio
Inicializar variables
color de fondo .... ND=1
inicialicemos la
primera columna .... X=0
Inicialicemos la
primera fila .... Y=0
color de tinta .... TIN=6
bandera (para levantar
el lápiz) .... NF=1
dimensionar las
matrices ... DIM PX(600), PY(600)
Preparar la pantalla
declarar el color de tinta
y el de fondo ..... COLOR TIN,1
declarar que usaremos
el modo de alta .... SCREEN 2
resolución para graficar

En la figura 13 mostramos como sería el esquema general del programa más conocido como esquema de bloques.

Con esto sí hemos terminado de explicar cada una de las sentencias del programa y hemos tratado de mostrar el planteo que se utiliza para crearlo.

Figura 13



Habilitar la barra de espacio y las teclas de función, inicializar las variables

Preparar la pantalla: color y definir que usaremos alta resolución

Definir el sprite (flecha)

Definir una línea para cada tecla de función

Operar para mover el sprite y dibujarlo

Pintar los puntos por donde pasamos, siempre y cuando NF=1

Volvemos a la parte que mueve el sprite

(Aquí se llega al presionar la barra)
Cambiar el estado de la bandera

(Se llega al presionar F1)
Cambiar el color de tinta
(Borrar puntos)

(Se llega por presionar F2)
Pinta cuadriláteros
(Borra regiones grandes)

(Se llega por presionar F3)
Guarda puntos claves

(Al presionar F4)
cortar continuidad del dibujo

(Por presionar F5)
Limpiar las matrices
(Borra cualquier dibujo almacenado anteriormente)

(Al presionar F6)
Reproduce el gráfico

MODO DE UTILIZAR EL PROGRAMA

Una vez verificado que esté correctamente copiado el listado del programa, debemos entrar la sentencia RUN. Ya sabemos cuales son las teclas que se utilizan para manejar el programa y para qué sirve cada una. Una vez terminado el dibujo que queríamos hacer, guardemos los puntos estratégicos. Previamente presionemos la tecla F4 para asegurarnos que no haya "basura" u otros dibujos guardados en las matrices. Al terminar de guardar los puntos más importantes del gráfico, debemos detener el programa presionando simultáneamente las teclas: CTRL y STOP. Luego entramos la siguiente línea:

1 PRINT A: PRINT: FOR F=1 TO A: PRINT PX(F): PRINT PY(F): NEXT

En la pantalla aparecerán los valores de las coordenadas. Pero como la impresión de estos valores se realizará muy rápido, aconsejamos ir deteniéndola presionando solamente la tecla STOP. Copiemos los números en el mismo orden que aparecen.

Una vez que tengamos estos datos, en el programa donde queremos copiar el dibujo, hay que agregar las siguientes sentencias:

100 READ A: FOR F=1 TO A: READ PX(F), PY(F): NEXT
110 DATA valor del primer número copiado, segundo número...

Los números de línea 100 y 110 los podemos modificar. Depende de dónde querramos colocar el dibujo de nuestro programa.

Luego de la orden DATA, deberemos colocar los números copiados antes, sin alterar el orden.

Y por último las sentencias:

120 PSET (PX(1), PY(1)),6
130 FOR F=2 TO A: IF PX(F)=-1 THEN PSET(PX(F+1), PY(F+1)),6: NEXT ELSE LINE -(PX(F),PY(F)), 6: NEXT

Estos números de líneas también son modificables, pero deberán seguir a las líneas anteriores.

Con esto, logrará copiar el dibujo en cualquier programa y de una forma bastante sencilla.

En el listado que mostramos hay variables seguidas por el signo de admiración. Esto es para que la máquina opere con mayor rapidez.

Lo único que faltaría es darle a cada sentencia que vimos en la explicación un número de línea. Esto lo mostramos con el listado.



VELOCIDAD

Para afirmar la rapidez de ejecución de un programa en código de máquina contra otro en Basic (que ambos lleguen al mismo resultado), les mostramos aquí un ejemplo para los que se resisten a aceptar esta desventaja de tiempos.

Con el listado de la figura 1, copiamos los caracteres en pantalla uno por uno, con un programa fundamentalmente en Basic.

El tiempo que tarda hasta terminar de ejecutar el programa es bastante.

Pero no es la única forma de copiar los caracteres sobre la pantalla. Por ejemplo, en la figura 2 proponemos una manera mucho más veloz que la anterior.

Este programa esta hecho básicamente sobre rutinas en códigos de máquina.

Se recurrió para esto a una rutina del sistema que comienza en la posición de memoria &h005C.

Esta rutina mueve bloques de memoria desde la RAM, hacia la VRAM. A los que entienden Assembler, les aclaramos que el par HL debe tener la dirección del bloque a transferir. En el par DE se guarda la dirección a partir de la cual se va a copiar el bloque, y el registro BC debe tener la longitud de éste.

Si ejecutamos cada programa, veremos la gran diferencia en tiempo que tarda entre uno y otro.

Los programadores en código de máquina pueden aprovechar la rutina del sistema mencionada, para incorporar en sus programas. Pero deben tener en cuenta los valores que mencionamos más arriba para los registros HL DE y BC.

Un INPUT diferente

Si te molesta el signo de interrogación que aparece después del INPUT, con las líneas de la figura 3 se solucionará.

Lo que hacen es colocar un parche sobre el signo "?". Luego de hacer correr esta rutina, verás las sentencias INPUT sin signo de pregunta. Para volver a visualizar estas sentencias con el signo, simplemente entra la instrucción:

POKE &HFDEO,&HC9

TRAMPA PARA BLAGGER

En el número 6 de nuestra publicación, comentamos un juego llamado BLAGGER. Los que lo probaron habrán notado lo difícil que es pasar de un nivel a otro.

Todos deseamos llegar al último nivel, pero escasi imposible pues, al perder las vidas, volvemos al inicio del juego. Con el listado de la figura 4, te mostramos una forma de hacer trampa. Se trata de convertirnos en inmortales o bien de poner nosotros mismos un límite de vidas.

De esta forma se nos amplian las posibilidades de llegar a la última pantalla, y ver qué sucede cuando logramos atravesar los diferentes desafíos. Simplemente debes copiar (y sin errores) el listado de la figura 4. Este programa se encargará de copiar el juego. No te olvides de grabar este programa cargador antes de hacerlo correr.

CODIGO DE CARACTERES

En la figura 5 mostramos un corto programa que, al ejecutarlo, imprimirá en pantalla los códigos ASCII de las teclas que vayas presionando. Como la mayoría de los manuales que vienen con las computadoras no traen apéndice con la lista de códigos y caracteres, con este programa se resolverá este inconveniente.

Figura 3

```
10 POKE &HFDE1,&HC3
20 POKE &HFDE2,&HD2
30 POKE &HFDE3,&H23
40 POKE &HFDEO,&HF1
```

Figura 1

```
10 COLOR 1,15,1:SCREEN 2
20 DEFINT A,C,D,X,Y
30 A=0:D=&H1BC7:C=2039
40 FOR Y=0 TO 2
50 IF Y=2 THEN A=4079: C=6120: D
=&H1BC7
60 FOR X=A TO C
70 VPOKE X,PEEK(D):D=D+1
80 NEXT X
90 IF Y=0 THEN A=2040: C=4079:D=
&H1BC7
100 NEXT Y
110 GOTO 110
```

Figura 2

```
10 COLOR 1,15:CLEAR 200,&HEFFF
20 SCREEN 2
30 FOR I=&HF000 TO &HF00C
40 READ A
50 POKE I,A
60 NEXT
70 DEFUSR=&HF000
80 GOSUB 160
90 POKE &HF004,&HF8
100 POKE &HF005,&H7
110 GOSUB 160
120 POKE &HF004,&HF0
130 POKE &HF005,&HF
140 GOSUB 160
150 GOTO 150
160 X=USR(1):RETURN
170 DATA &h21,&hc7,&h1b,&h11,&h0
,&h0,&h1
180 DATA &hf8,&h7,&hcd,&h5c,&h0,
&hc9
```

Figura 4

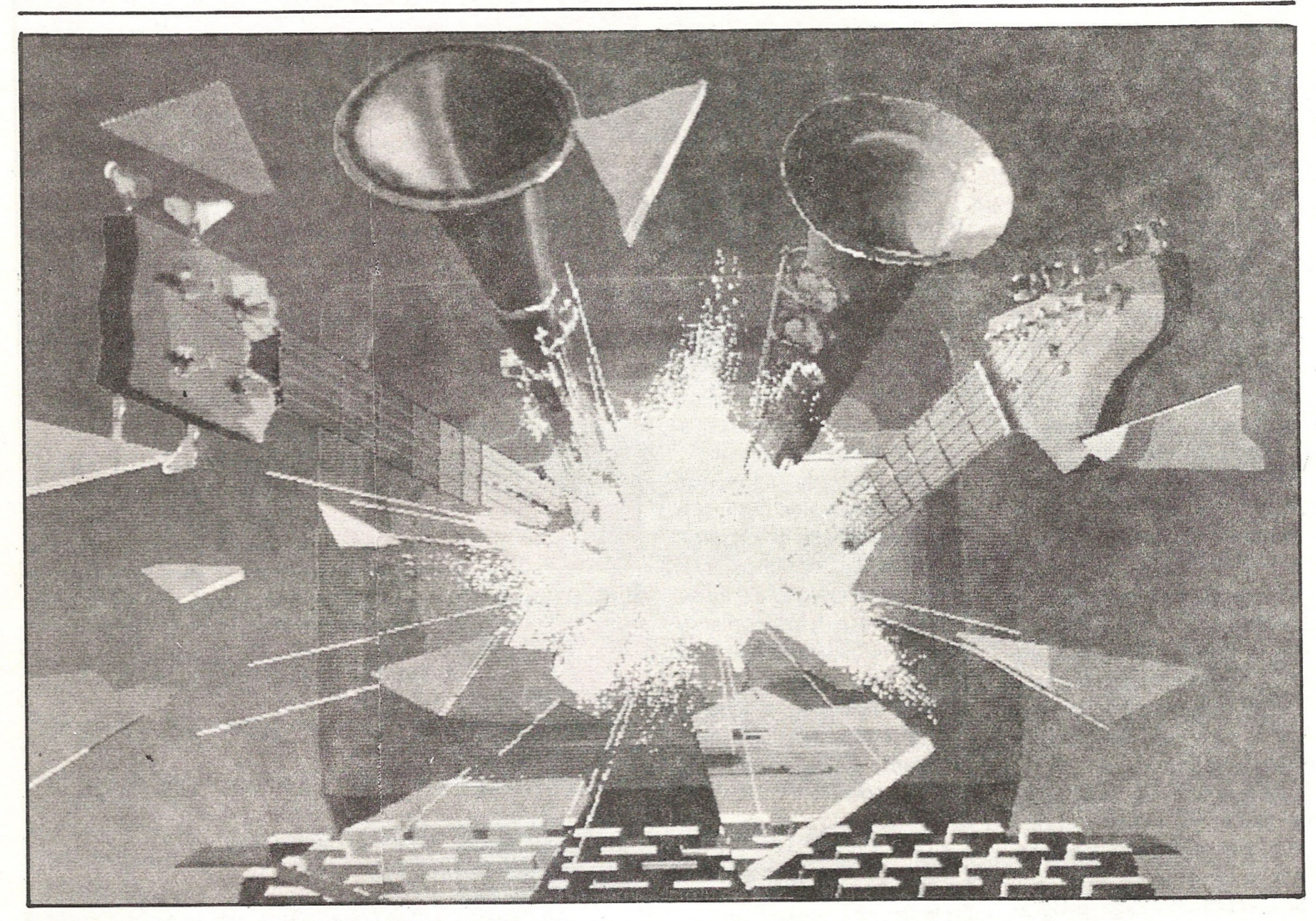
```
10 KEY OFF:CLS
20 PRINT"Querés no tener limite
de vidas? (S/N)"
30 Z$=INPUT$(1):IF Z$="n" OR Z$=
"N" THEN 70
40 IF Z$="s" OR Z$="S" THEN 50 E
LSE 30
50 D1=&H9C3D:D2=&H9CDD
60 DT=201:GOTO 100
70 CLS:INPUT"Número de vidas";DT
80 IF DT>60 THEN 70
90 D1=&H9254:D2=D1
100 SCREEN 1,0,0,1:BLOAD"cas:",R
:BLOAD"cas:"
110 DEFUSR=37202!:POKE D1,DT:POK
E D2,DT:A=USR(0)
```

Figura 5

```
10 CLS:KEY OFF:SCREEN 0:COLOR 1,
4
20 E$=INKEY$:IF E$="" THEN 20
30 FOR I=0 TO 255
40 IF E$=CHR$(I) THEN PRINT"El c
ódigo decimal de"; CHR$(I);:PRINT
"es: ";I
50 NEXT
60 GOTO 20
```

DESNUDAMOS EL CHIP DE SONIDO

Veremos las funciones de cada patita y así podremos entender qué sucede cuando creamos música desde el Basic. Analizaremos los procesos internos desde una óptica algo técnica.



El culpable de los sonidos en nuestro computador es el chip denominado AY-3-8910. Es conocido también por las siglas PSG (Programable Sound Generator).

El Ay-3-8910 tiene varias patitas, cada una con un nombre, como se ve en la figura 1.

Cada pata tiene una función. Pero reemplazaremos la palabra función por ''registros''.

Cada registro va acompañado por un dato. De esta forma se le indica al PSG lo que debe hacer.

Pero el PSG no es el chip maestro de la computadora. O sea la CPU (cuyo nombre es Z80) funciona por separado, aunque existe una constante comunicación entre ambos.

El Z80 maneja al chip de sonido como si fuera otro periférico, como podría ser una imperesora o un joystick.

Veamos un poco la función de cada patita del Ay-3-8910:

* **IOA0-IOA7, IOB0-IOB7:** son líneas de entrada y salida A y B, conocidas como puerto de entrada-salida.

A estas líneas es donde se conectan los periféricos. Cada accesorio (como los joysticks) debe dejar y leer señales para poder funcionar. Pero no todos los periféricos leen en el mismo tiempo que el procesador manda la señal de orden, ni pueden ser todos conectados al mismo ''bus de datos'', ni entienden todos la información en el sistema binario. Una forma de solucionar estos inconvenientes y muchos más, fue la creación de estos puertos de entrada-salida (E/S).

Imaginemos a estos puertos como lugares donde los datos son alojados por el procesador hasta que puedan ser leídos por el periférico correspondiente, o hasta que se conviertan para que puedan ser entendidos por este último. Consisten en líneas conectadas al bus de datos del Z80 y otra al bus de datos del periférico. Los dos puertos para el PSG son *IOA0-IOA7* llamado puerto externo A y *IOB0-IOB7*: puerto externo B. Los datos son leídos y escritos en estos

registros (R14 para A y R15 para B). Es conveniente destacar, para que quede claro, que la única tarea de estos registros es la de entrar o sacar datos.

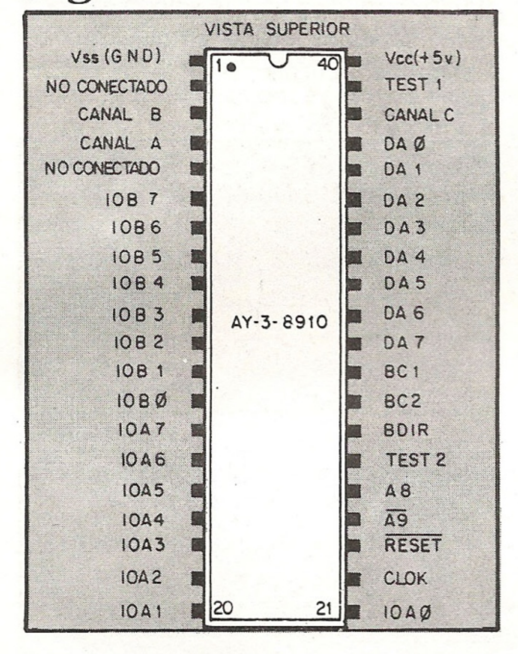
* **DA0-DA7:** es otro bus de datos, pero sirve para comunicarse con el micro y seleccionar el registro a modificar.

* **A8-A9:** añaden 2 bits de significación al PSG para aquellos sistemas que tengan conectados más de un periférico. Si por ejemplo deseamos seleccionar el funcionamiento del PSG, deberemos darle a A8, +5 voltios, o sea, nivel lógico 1, ya la patita A9: 0 voltios, es decir, nivel lógico 0.

Si damos cualquier otra combinación no reconocerá el funcionamiento de DA0-DA7. En consecuencia no podremos modificar ningún registro.

* **BDIR, BC2, BC1:** son las encargadas de decodificar, traducir, las señales enviadas en DA0-DA7, tanto para la entrada como para la salida de datos. Una combinación entre las tres líneas hace que se puedan ejecutar varias operaciones. La forma de conseguirlo está detallada en la figura 2.

Figura 1



* **TEST 1 - TEST 2:** no cumplen ninguna función aprovechable para el usuario. Fueron creadas por el fabricante para testear el buen funcionamiento del chip (para tener una garantía de que funcionarán luego de salir de su fábrica, en una brillante MSX armada). Es aconsejable no conectarle nada a estas patitas porque no se tiene abundante información sobre ellas.

* **CLOCK:** es una entrada de tipo TTL, y se usa para hacer los tiempos de los tonos, ruidos y envolventes.

* **VCC, VSS:** alimentación de tensión para el chip. Debe ir +5 y 0 voltios en VCC y VSS respectivamente.

* **Canal A, B y C:** salida analógica del sonido para cada canal: A, B y C.

ESCRIBIR Y LEER REGISTROS

Para escribir en registros del PSG se deberá: 1) colocar el dato a escribir en DA0-DA7; 2) aclarar que escribiremos una dirección, con las combinaciones adecuadas de las patitas BDIR, BC1 y BC2 que generalmente se utiliza (ver figura 2) BDIR=1, BC1=1; 3) y el par BC1 y BC2=0 para dar un valor a un registro.

En cambio para leer un valor deberemos 1) darle a BDIR=0, BC1=1 y BC2=1 para después leer el dato en DA0-DA7. O sea que para escribir un valor en un registro será: 1) escribir con BC1=1 y en DA0-DA3 representar en binario el valor del registro a modificar; 2) escribir con BC1=0 y en DA0-DA7 colocar el valor a pasar al registro.

Y para leer un valor de un registro: 1) igual que para escribir un valor en un registro y 2) realizar una operación de lectura, leyendo en DAC DA7 el valor del registro indicado.

CREANDO TONOS

Se utilizan 2 contadores: uno se inicializa con el contenido del registro de control del canal correspondiente. Los registros para cada canal son R0 y R1 para el A, R2 y R3 para el B1 y R4, R5 para el C. El número con que se inicializará este contador es de 12bits: 8 para ajuste fino y 4 (bit 0-bit 3) para el grueso

Este valor inicial irá decreciendo en 1 por cada 2 ciclos el clock. Al llegar a cero, vuelve a tomar otro valor del registro de control de algún canal y genera

Figura 2

BDIR	BC1	BC2	FUNCION
0	0	0	INACTIVO
0	0	1	ESCRIBE DIRECCION
0	1	0	INACTIVO
0	1	1	LEE DATO
1	0	0	ESCRIBE DIRECCION
1	0	1	INACTIVO
1	1	0	ESCRIBE DATO
1	1	1	ESCRIBE DIRECCION

un pulso. Este pulso sirve de entrada para el segundo contador. Puede ser sólo un número de 2 bits, o sea cualquiera de estos valores: 00, 01, 10 y 11.

Cada vez que recibe un pulso del primer contador, aumenta en uno su valor hasta llegar a 11. Luego vuelve a repetir este valor (11) para después comenzar a restar 1 también con la llegada de cada pulso. Cuando llegue a 00, repite este valor nuevamente y otra vez le suma 1 hasta llegar a 11. Este ciclo se repite sucesivamente hasta que el registro de control de tono se ponga en 0.

¿Pero qué sucedería si tomáramos los 2 bits del segundo contador y lo utilizáramos como entrada de un conversor analógico digital? Pues sí, obtendríamos como respuesta una señal periódica en el tiempo, con una frecuencia determinada por la fórmula:

$$f = \frac{fr}{16 * tp}$$

donde f es la frecuencia de la señal de salida y fr es 1.7897725 MHZ para las MSX (frecuencia del reloj interno) y tp es el contenido del registro de control de tono.

Veamos un ejemplo: ¿cómo sería para obtener una frecuencia de 440 Hz, en las MSX? Según vimos, utilizando la fórmula sería poner:

$$\frac{1.7897725}{16 * 440} = 254$$

en el registro de control A (puede ser otro canal si quisiéramos). Este número se repartiría en 254 para el R0 y 0 para R1.
Las figuras 3, 4 y 5 son para los que entienden un poco más de electrónica y desean un estudio más profundo sobre la generación de tonos.

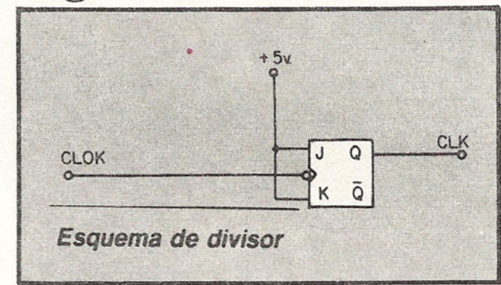
Figura 3

Esquema del bloque base.

CREANDO RUIDOS

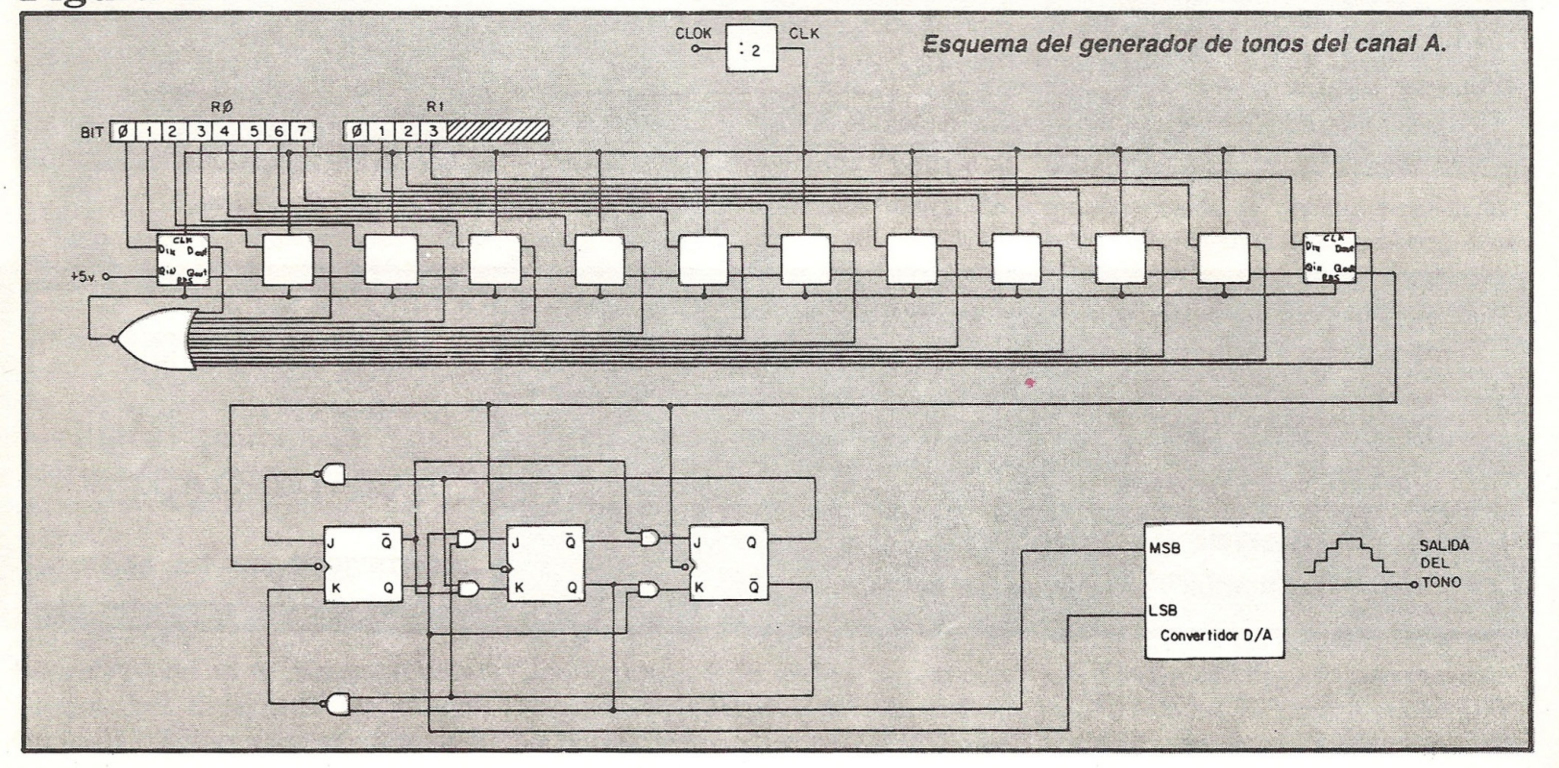
La forma de generarlos es similar a la usada para los tonos, salvo que el contador se inicializó con el valor del registro R6 en los bits 0 o 4 poniéndose los 7 bits superiores en cero. Con estos datos se genera una señal que modula la amplitud del generador de ruido blanco y suena verdaderamente como si fuera un ruido. Si bien hay un sólo canal para esto, debemos indicar en qué canal lo deseamos escuchar, si en A, B o C.

Figura 4



Esquema de divisor

Figura 5



Esquema del generador de tonos del canal A.



EL COMPOSITOR AUDAZ

CLASE: UTILITARIO

Yo se trata de un programa insólito y exclusivo. Esta es una versión diferente de los simuladores de piano que permitirá crear melodías.

Contamos sólo con una octava, pero tenemos la ventaja de poder modificar el tono y la duración.

Al dar una primera vista al listado, notaremos que posee estructuras con POKEs y PEEKs. Esto dificulta un poco el seguimiento para los que quieran entender detalladamente cómo es el funcionamiento de este utilitario. Pero gana velocidad cuando se ejecuta el programa.

Para variar el tono, se utiliza la tecla "1". Al presionarla, irá cambiando el tono, con la correspondiente identificación. Es decir, que simultáneamente al teclear "1", se visualizará en pantalla el cambio de tono y auditivamente notaremos la modificación. Podemos elegir entre tres tonos distintos, y se cambia la elección solamente presionando la tecla "1". Recordalo. En cuanto a la duración, con la tecla "2" disminuiremos la duración y con la "3" aumentaremos.

Lo más importante que diferencia este programa de los otros utilitarios del mismo estilo, es la rapidez entre la respuesta del tono y la tecla presionada.

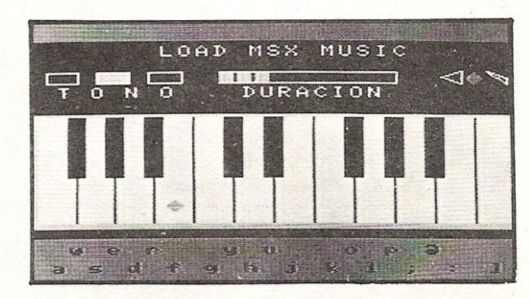
Debajo del dibujo del piano están escritas las teclas de la computadora que corresponden a cada tecla del piano.

Las teclas de función no están desactivadas y ésta es una gran ventaja. Veamos porqué.

Si "F5" está definida con la sentencia RUN, el programa hará sonar las notas correspondientes a la "R" y a la "U". La "N" no está definida. Entonces, si definimos "F1" como "asdfghj", hará sonar una escala ascendente, correspondiente al tono de cada letra.

Les daremos una sugerencia de cómo aprovechar esta ventaja: primero practiquen bien una melodía y memoricen el orden de las letras. Luego definan las teclas de función con las mismas letras (y en el orden adecuado) que lleva la música.

Cuando vuelvan a ejecutar el programa y presionen las teclas de función, oirán la música que han creado. Interesante, ¿no?



VARIABLES IMPORTANTES:

PX, PY: matríz con las coordenadas del sprite guía.

TP: matriz con los tonos destinados a cada tecla.

ESTRUCTURA DEL PROGRAMA:

10-140: inicializa variables, lee datos del DATA.

150-280: dibuja la pantalla.

290: lee la tecla presionada.

300: ubica el sprite guía sobre el piano según la tecla presionada.

310-330: toca el tono correspondiente.

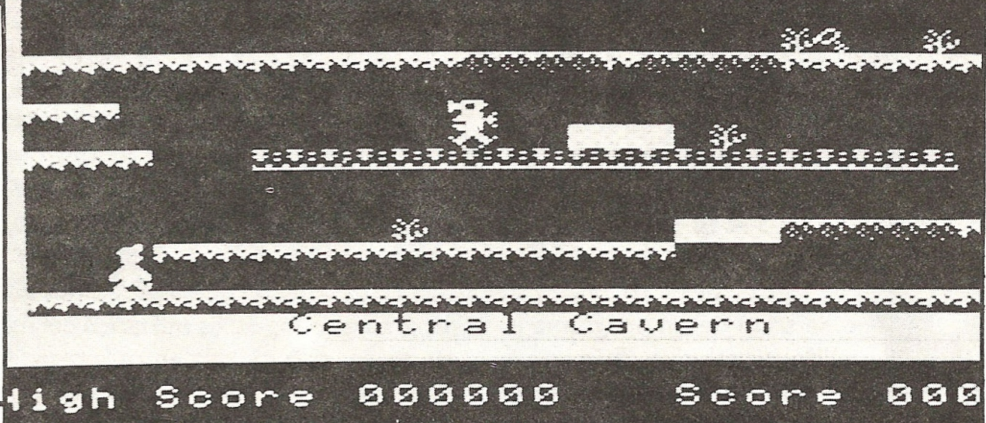
340-530: datos.

540-590: controla el tono.

600-700: controla la duración.

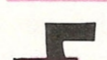
```
10 CLEAR 300,&HD7FF:S=0:FOR N=&H
D800 TO &HD857
20 READ H$:X=VAL("&H"+H$):S=S+X:
POKE N,X:NEXT
30 IF S<>7686 THEN PRINT"No esta
n bien los valores de las lineas
340-530"
40 DIM TN(11),TP(45),PX(45),PY(4
5),LX(45),LY(45),KY(20):DEFUSR0=&
HD800
50 POKE&HFCAB,0:TQ=-1:RT=5:FOR N
=0 TO 11:READTN(N)
60 IF TN(N)=1 THEN PY(N+7)=110 E
LSE PY(N+7)=135
70 NEXT:PX(19)=224:PY(19)=135
80 FOR N=0 TO 45:LX(N)=214:LY(N)
=49:READ TP(N):NEXT
90 FOR N=7 TO 18:READ PX(N):PX(N
)=PX(N)-5:NEXT
100 FOR N=0 TO 6:PX(N)=PX(N+12)-
125:PY(N)=PY(N+12):NEXT
110 FOR N=20 TO 25:LX(N)=234:PX(
N)=PX(N-13):PY(N)=PY(N-13):NEXT
120 FOR N=26 TO 31:LX(N)=194:PX(
N)=PX(N-19):PY(N)=PY(N-19):NEXT
130 FOR N=32 TO 38:LX(N)=234:PX(
N)=PX(N-19):PY(N)=PY(N-19):NEXT
140 FOR N=39 TO 45:LX(N)=194:PX(
N)=PX(N-39):PY(N)=PY(N-39):NEXT
150 COLOR 15,12,12:SCREEN 2:OPEN
 "GRP:" FOR OUTPUT AS#1
160 SPRITE$(0)=CHR$(&H18)+CHR$(&
H3C)+CHR$(&H7E)+CHR$(&HFF)+CHR$(&
HFF)+CHR$(&H7E)+CHR$(&H3C)+CHR$(&
H18)
170 LINE(0,30)-(255,160),6,BF:LI
NE(10,30)-(246,160),1,BF
180 LINE(16,78)-(238,154),14,BF:
LINE(20,80)-(236,150),15,BF
190 PSET(30,33),1:PRINT#1,"
 LOAD MSX MUSIC"
200 FOR N=0 TO 10:X=N*18+38:LINE
(X,80)-(X,150),1
210 READ I:IF I=1 THEN LINE(X-4,
80)-(X+4,120),1,BF
220 NEXT:GOSUB 540:GOSUB 630
230 LINE(203,53)-(212,49),15:LIN
E-(212,57),15:LINE-(203,53),15
240 LINE(232,53)-(223,49),15:LIN
E-(232,57),15:LINE-(232,53),15
250 PSET(30,60),1:PRINT#1,"T O N
 O"
260 PSET(115,60),1:PRINT#1,"DURA
CION"
270 COLOR 1:FOR N=0 TO 19:PSET(P
X(N),(PY(N)-110)/2+165),12:PRINT#
1,CHR$(PEEK(&HD829+N)):NEXT
280 SOUND 7,&B00111110:SOUND 8,1
6:SOUND 9,16:SOUND 12,RT*10:SOUND
 13,0
290 POKE&HF3F7,0:L=USR0(0):IF L=
-1 THEN GOSUB 670:GOTO 290
300 PUT SPRITE 1,(LX(L),LY(L)),4
,0:PUT SPRITE 0,(PX(L),PY(L)),9,0
:ON TQ+1 GOTO 330,320,310
310 SOUND 2,(TP(L) MOD256)+1:SOU
ND 3,TP(L)/256:GOTO 330
320 SOUND 2,(TP(L)/2) MOD256:SOU
ND 3,TP(L)/512
330 SOUND 0,TP(L) MOD 256:SOUND
1,TP(L)/256:SOUND 13,0:GOTO 290
340 DATA CD,9F,00,1E,00,21,29,D8
350 DATA 01,2F,00,ED,A1,E2,19,D8
360 DATA 28,03,1C,18,F6,16,00,18
370 DATA 03,11,FF,FF,32,5F,D8,3E
380 DATA 02,32,63,F6,ED,53,F8,F7
390 DATA C9,61,77,73,65,64,72,66
400 DATA 67,79,68,75,6A,6B,6F,6C
410 DATA 70,3B,40,3A,5D,47,59,48
420 DATA 55,4A,4B,07,19,08,15,0A
430 DATA 0B,4F,4C,50,2B,60,2A,7D
440 DATA 01,17,13,05,04,12,06,a0
450 DATA 0,1,0,1,0,0,1,0,1,0,1,0
460 DATA 320,302,285,269,254,240
470 DATA 214,202,190,180,170,160
,151,143,135,127,120,113,107
480 DATA 107,101,95,90,85,80
490 DATA 428,404,381,360,339,320
500 DATA 75,72,68,63,60,56,53
510 DATA 640,604,570,538,508,480
,452
520 DATA 102,111,120,129,138,156
,165,174,183,192,201,210
530 DATA 1,1,1,0,1,1,0,1,1,1,0
540 TQ=(TQ+1) MOD 3
550 IF TQ>=1 THEN SOUND 7,&B0011
1100 ELSE SOUND 7,&B00111110
560 FOR T=0 TO 2:LINE(T*24+24,50
)-(T*24+39,57),1,BF
570 LINE(T*24+24,50)-(T*24+39,57
),15,B:NEXT
580 LINE(TQ*24+25,51)-(TQ*24+38,
56),9,BF
590 RETURN
600 RT=RT+1:IF RT>10 THEN RT=10
610 GOTO 630
620 RT=RT-1:IF RT<1 THEN RT=1
630 LINE(104,50)-(183,57),1,BF
640 LINE(104,50)-(183,57),15,B
650 FOR T=1 TO RT:LINE(96+T*8,51
)-(102+T*8,56),3,BF:NEXT
660 SOUND 12,RT*10:RETURN
670 CM=PEEK(&HD85F):IF CM=&H31 T
HEN 540:RETURN
680 IF CM=&H33 THEN 600:RETURN
690 IF CM=&H32 THEN 620:RETURN
700 RETURN
```





MANIC MINER

CREATIVIDAD: ***8***
PRESENTACION: ***10***
ATRACCION: ***8***
SONIDO: ***9***
GRAFICOS: ***9***
TIPO: ***ENTRETENIMIENTO***
PRODUCE: ***BITGAME***

on este juego tenemos la oportunidad de convertirnos en un minero que recorrerá una serie de cavernas.

De acuerdo a nuestra lógica, un minero busca oro o un tesoro, pero este es diferente. Su ambición consiste en recolectar todas las llaves distribuidas en las diferentes cuevas.

Cada caverna tiene su trampa, pero todo se soluciona cuando desciframos el truco que nos impide recoger a cada llave.

Durante todo el juego nuestros oidos percibirán una melodía agradable y suave, pero que conseguirá aumentar nuestros nervios por la frecuencia musical en la que está creada.

Manic Miner es un entretenimiento muy bien pensado y cuenta con una amplia cantidad de gráficos como teléfonos, pingüinos, patos y focas de circo. La pantalla inicial está representada por una granja con una casita, un perfecto árbol y una laguna donde se refleja un sol muy especial.

Pero, además de la excelente graficación de este cuadro, esta pantalla se destaca por la presencia de un teclado musical.

La música principal de Manic Miner se escucha sólo en la pantalla inicial, y en el piano se marcan las teclas que se presionan para crear esta melodía.

Este pasatiempo es del mismo estilo que BLAGGER, ya comentado en el número 6 de esta revista.

En definitiva, un juego de mucha acción, donde no sólo los reflejos son importantes, sino que la astucia para encontrar el mejor camino, es fundamental.

SPOOKS & LADDERS

CREATIVIDAD: ***8***
PRESENTACION: ***7***
ATRACCION: ***7***
SONIDO: ***6***
GRAFICOS: ***8***
TIPO: ***ENTRETENIMIENTO***
PRODUCE: ***BITGAME***

Para juegos de acción, Spooks & Ladders es ideal. se trata de subir y bajar escaleras, esquivar fantasmas y recoger elementos valiosos como llaves, plata y joyas.

No es sencillo cumplir el objetivo de recolectar los objetos, son pocas las escaleras por las que tendremos acceso a los diferentes pisos.

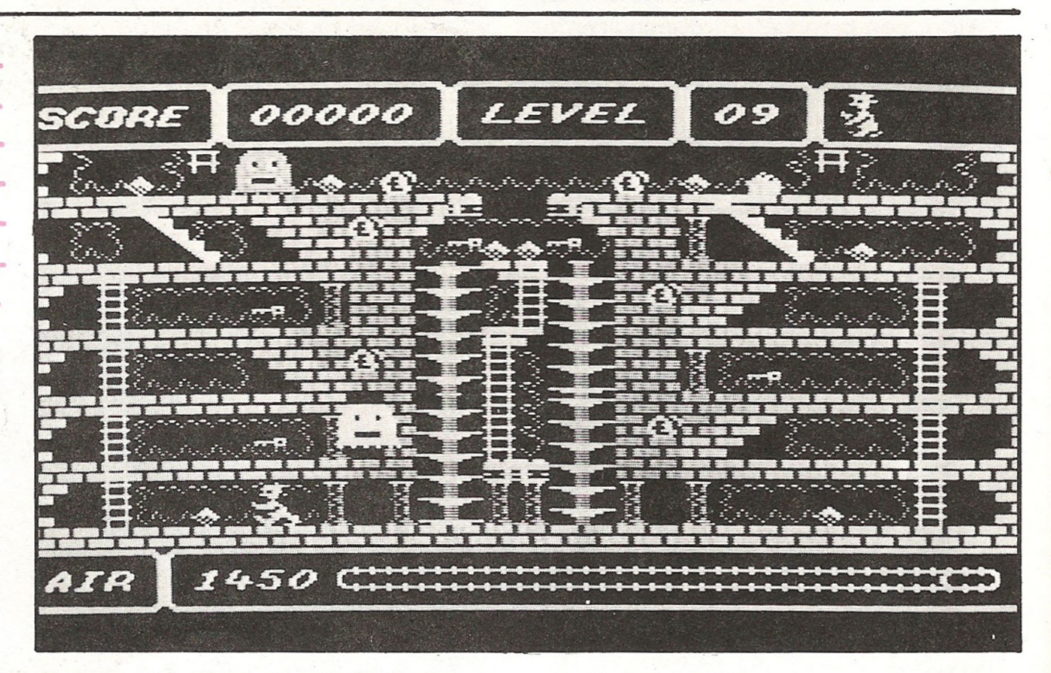
Si alcanzamos escapar por las salidas de aire ubicadas en la parte superior de la pantalla, obtendremos puntos extra.

Para los carteles y comentarios utilizados en este juego, se redefinió el set de caracteres.

Son varias las pantallas que forman la travesía de Spooks & Ladders.

Una ventaja de este entretenimiento es que nosotros podemos elegir el nivel de dificultad con el cual queremos comenzar.

Como podemos ver en una de las primeras pantallas aclaratorias, cada objeto tiene distinto puntaje, pero están dados por la dificultad que representa tomarlos.



Nos encontramos frente a un juego muy atrapante, cualidad esencial de estos programas comerciales, debido a que durante el juego con Spooks encontrarás constantemente distintas sorpresas.

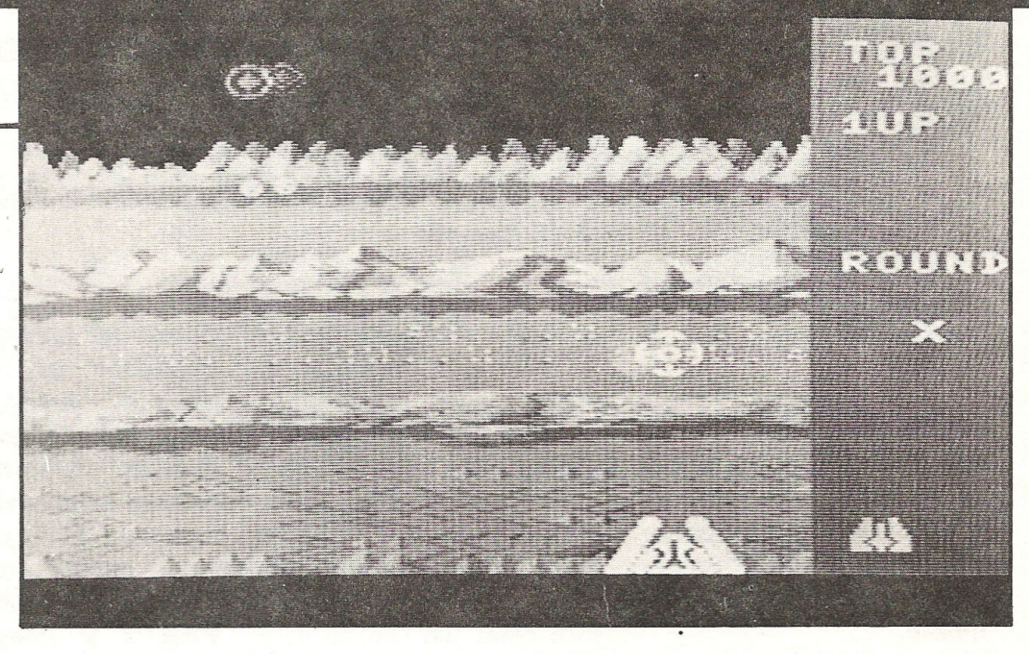
De acuerdo a tu habilidad, pasarás a incorporar la tabla de honor de Spooks & Ladders, formada solo por los seis mejores jugadores.

Una vez cargado el programa, si presionas una sola vez la barra de espacio, verás todas las pantallas aclaratorias con las explicaciones del juego, los puntajes de los objetos, etc. Pero necesitarás tener paciencia.

Y para concluir este comentario, queremos dirigirnos a los que aún no conocen el juego, dos detalles: 1° los fantasmas mueven los ojos como si nos estuvieran tratando de encontrar con la mirada para perseguirnos, y 2° cuando seamos alcanzados por los fantasmas, nuestro cuerpo se transformará y consumirá de una forma particular y graciosa al mismo tiempo.

SENJYO

CREATIVIDAD: **5**
PRESENTACION: **6**
ATRACCION: **8**
SONIDO: **8**
GRAFICOS: **7**
TIPO: **ENTRETENIMIENTO**
PRODUCE: **BITGAME**



Hagamos volar un poco nuestra imaginación y pensemos cómo puede llegar a estar distribuida la vida dentro de este infinito cuerpo celeste donde vivimos.

Tal vez no seamos los únicos seres vivientes. Tal vez, en la Tierra no sean solamente "hombres" los habitantes. Tal vez el paisaje de la capa terrestre no sea como la de estos días.

En fin, en el futuro se pueden producir varios cambios.

SENJYO tomó una pequeña versión de lo que podría pasar después de varios siglos en nuestro Planeta.

Se podría interpretar como una versión un poco pesimista sobre el destino de la Tierra, pues seremos atacados continuamente por naves extraterrestres. Estas naves dispararán sucesivamente contra nosotros con el fin de destruirnos.

Nuestro objetivo es defender nuestro terreno sin ser destruidos. El paisaje futurista que propone este entretenimiento consiste en un cielo oscuro, pacífico y limpio, que inspira sensación de infinito. Las montañas elevadas y punteagudas forman barreras en la superficie de este extraño paisaje.

No se ven vidas aparte de los extraterrestres. Ni un simple animal o vegetal se incorpora al grupo de protagonistas de este juego.

Las naves enemigas aparecerán desde el horizonte y se desplazarán por la superficie como acariciando las laderas de las montañas.

Todo parece muy pacífico, pero esta no es una cualidad que destaca a SENJYO. Este es un juego de mucha acción donde los reflejos son importantes.

El rápido avance de las naves atacantes activan los nervios de cualquier jugador, incluso de los más serenos. Es conveniente no quedarnos quietos, así impediremos que nos puedan apuntar.

La calma del cielo se romperá cuando pasen (sólo en determinados momentos) extraños objetos voladores a los que deberemos destruir.

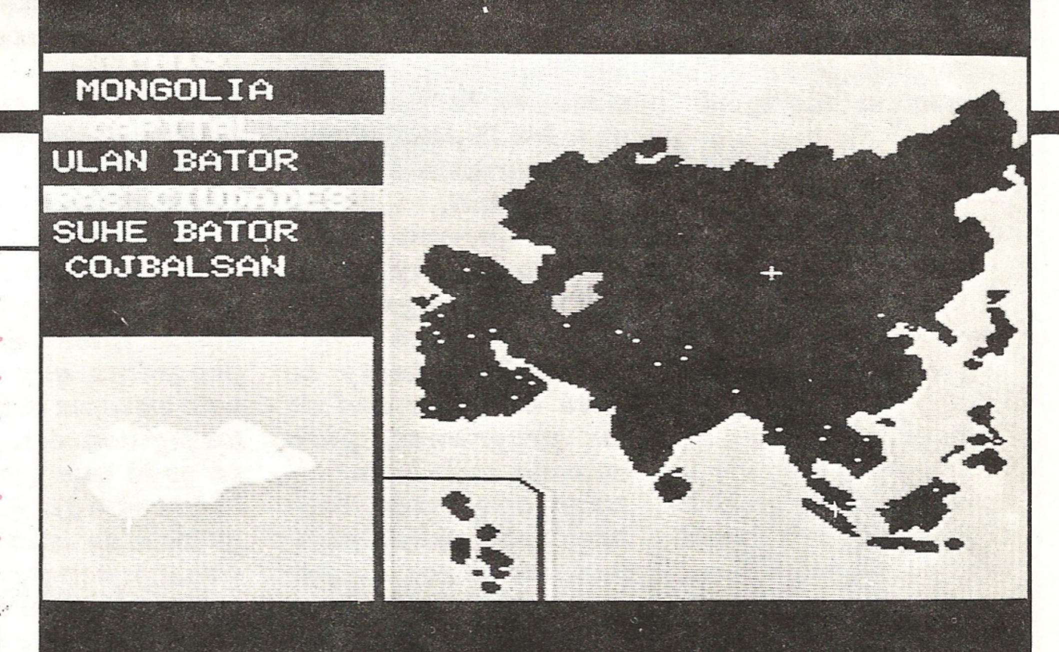
Muy original es la forma de anotar los puntos. Consiste en una tabla cuadriculada con un total de 30 casilleros. Al destruir las naves llenamos uno de estos cuadraditos, en cambio si derribamos a los objetos voladores, seremos más afortunados rellenando dos casilleros. Cuando logremos completar la tabla, pasaremos de nivel.

La dificultad de los niveles crece por ser cada vez mayor el número de naves que nos atacan. Y para ser más interesante el desarrollo, cambian las figuras de los objetos voladores.

Esta es una versión diferente de los clásicos juegos espaciales que fueron el origen de los entretenimientos electrónicos y tiene la ventaja de ser fácil de manejar.

ASIA

CREATIVIDAD: **6**
PRESENTACION: **7**
ATRACCION: **7**
SONIDO: **NO**
GRAFICOS: **8**
TIPO: **EDUCATIVO**
PRODUCE: **BITGAME**



Parte de nuestra cultura consiste en conocer las ubicaciones y detalles importantes de los países que constituyen el globo terráqueo.

Algunos continentes son sencillos de estudiar como Europa y América. Pero el objetivo se complica cuando se trata de los continentes africano y asiático.

Una forma más agradable de memorizar la parte geopolítica de los países orientales, es recurrir a este programa llamado ASIA.

Consiste en una dinámica lección con información sobre los países, como por ejemplo su religión, población, moneda y lengua.

Moviéndonos por el continente con los cursores, podemos elegir el país sobre el cual necesitamos información.

Con la barra de espacio aparecerá el nombre del mismo, luego al presionar RETURN obtendremos información. Y si posteriormente presionamos la barra espaciadora, se imprimirán sobre la pantalla, toda la información restante del país elegido.

El contorno del continente y de los países está perfectamente logrado, y su calidad como programa no es perjudicada por no contar con sonido.

Insistimos que se trata de un educativo muy didáctico, que permitirá al usuario incorporar conocimientos en forma divertida.



Para comunicarse con nosotros deben escribirnos a "Load Revista para Usuarios de MSX", Paraná 720, 5to. Piso, (1017) Cap. Federal.

RESET

Me gustaría que expliquen si esta computadora posee "ventanas" y qué otros periféricos vendrán próximamente.
Lo que más interesa es que hagan un "reset" para ponerlo dentro de la computadora. Comenten dónde puede ubicarse para no tocar otros elementos, cuáles son las piezas necesarias para construirlo y cómo se piden en las casas de electrónica. ¿Puede dejarse permanentemente conectado a la computadora por medio de una soldadura y así tener cualquier otro periférico conectado normalmente?
¿El idioma Pascal viene en cassette? Los felicito por la revista que es muy buena y porque todo lo que trae es entretenido. Una parte interesante es "Raiting Soft".

Alejandro Furno - Santa Fe

LOAD MSX

La existencias de ventanas depende del programa, algunos programas las incorporan.
Todas las novedades que aparecerán en el mercado las comentamos en la sección FILE. Y algunos de los periféricos que saldrán son realmente interesantes.
No hace falta construirse un reset para la Talent: está incorporado. Sobre las teclas SELECT y HOME, hay un rectángulo amarillo con las letras MSX en negro. Este rectángulo está apoyado en una tapa que se hunde. Si hundimos esta tapa a la altura de la tecla INS notaremos que hay una chapita plateada. Al tocar esta chapita en su punto medio, provocamos un reset de la computadora. Este tipo de reset lo podemos utilizar con confianza, pero debemos evitar que no entre ningún elemento extraño al hundir la tapa.
Este reset fue creado para que al insertar los cartuchos se produzca el reset adecuado para no dañar la computadora.
Con esto se evita colocar los cartuchos con la máquina encendida (el reset provoca una interrupción de corriente pequeña como para apagar la máquina). Eso podría provocar daños en el cartucho y en la máquina. Con respecto a Pascal, si viene en cassette.

PROBLEMA EN LOS PROGRAMAS

Llegó a la editorial una carta del Dr. Norberto W. Melhem, planteándonos varias inquietudes. No la publicamos textualmente debido a su longitud.
Antes, queremos aclarar que nuestra crítica de "El libro gigante de los juegos para MSX" de Andrew Lacey se basó en la calidad de los programas, pero esto no incluía garantizar que estén bien impresos en sus páginas. Nosotros no hemos copiado ningún programa, solo los juzgamos estudiando los listados. Nos pregunta sobre algunos problemas que tiene con los programas de este libro.
No encontramos a simple vista cuál es el error. Necesitamos un poco de tiempo para estudiar el inconveniente. Además nos dice "Duda 2: El programa "verificador" que se encuentra en el mismo libro, pide que se lo grabe con la orden SAVE"CAS:AUTVER"; y luego pide que se lo cargue en la computadora con la orden LOAD"CAS:AUTVER".
En las instrucciones del Manual de la computadora, dice que las grabaciones deben efectuarse con la orden "CSAVE"XXXXXX" y cargarse con CLOAD"XXXXXX" debiendo haber entre comillas hasta 6 caracteres. Entonces habría que aclarar este punto aparentemente contradictorio.
Duda 3: ¿Qué significa programa ASSEMBLER o en ASSEMBLER?
Duda 4: En la revista MSX 3, de la página 5, "Sprites Vivos", en donde dicen: "las primeras líneas de todo programa que utilice este sistema deben ser las siguientes:..." (esto es lo que me llama la atención), veo que al lado de las sentencias u órdenes, a la derecha, encolumnadas se lee una serie de sentencias BASIC y a continuación otras como LDHL,6912.
Preganto: ¿qué significa y por qué van encolumnadas a la derecha y alejadas del número de línea y no a continuación? ¿Cómo se transcribe ésto en la computadora?"

LOAD MSX

Sobre la duda 2, hay dos formas de grabar. Una con CSAVE-CLOAD, y de esta manera los archivos se graban en formato binario comprimido. Esto significa que se guardan tokens (códigos claves que identifican a las sentencias) en forma binaria y una a continuación del otro. En cambio, SAVE-LOAD carga o graba los archivos en forma de ASCII. Esta manera es muy lenta, porque en vez de guardar la sentencia, por ejemplo, PRINT bajo su código correspondiente (token), lo desarma en caracter por caracter y guarda los códigos ASCII correspondientes a la P,R,I, etc.
Pero la ventaja de tener archivos grabados en formato ASCII (con SAVE), es que estos archivos se pueden mezclar con otros sin importar en el formato que se encuentren grabados.
La mezcla entre programas se realiza mediante el MERGE. Para utilizar esta sentencia es recomendable leer su explicación en el manual de la computadora.
La duda 3 es muy sencilla contestarla. Primero deberíamos aclarar qué significa Assembler. Este es el lenguaje directo que entiende la computadora. Consiste en una serie de números, cada uno con un significado y que deben unirse de una forma en particular. Se puede escribir programas en este lenguaje.
Y sobre la publicación de la página 5, no se pudo hacer de otra forma más clara. La primera columna son las sentencias BASIC, y en la segunda columna está el significado del programa en Assembler cargado (en código) dentro del DATA.
Sólo hace falta cargar las sentencias BASIC. En las páginas de la revista lamentablemente se ven un poco apretadas las dos columnas y esto confundió.

Computación, una oportunidad para que todos enseñen y aprendan.

Un lugar para

desarrollar el pensamiento.

descubrir una vocación.

manejar lenguajes de computación.

comprender los múltiples usos de un computador.

capacitar y perfeccionar al docente.

incorporar los avances tecnológicos.

que el profesional domine el uso de nuevas herramientas.

que los padres se reencuentren con sus hijos.

"No se trata solamente de adquirir en forma puntual conocimientos definitivos, sino prepararse a elaborar a lo largo de toda la vida, un saber en constante evolución y de aprender a ser."

UNESCO

Actividades '86

Para Niños, Adolescentes, Adultos, Docentes, Profesionales y Establecimientos educativos.

INTRODUCCION A MICROCOMPUTADORES

DIAGRAMACION ESTRUCTURADA

LOGO

BASIC

COLOR - SPRITE - SONIDO

COBOL

PASCAL

ASSEMBLER

MS - DOS Y MSX - DOS

D BASE II - MULTIPLAN

PROCESADOR DE LA PALABRA

INSTALACION DE LABORATORIOS

en Establecimientos educativos con formación de multiplicadores y apoyo a la comunidad.

Cómo?

- Taller en grupos de 12 a 15 personas.
- Clases de 2 horas diarias.
- 2 ó 3 alumnos por equipo.
- Equipos disponibles para prácticas adicionales en horarios libres.
- Becas rentadas en el Departamento de investigación y desarrollo de Talent MSX.
- Becas rentadas para docentes en Laboratorios de Establecimientos Educativos.

Informes, Inscripción y Cursos

Lunes a Viernes de 8 a 22 hs.
Sábados de 8 a 13 hs.
CENTRAL:
Cabildo 2027 - 1er. Piso "A" (1428)
Capital Federal
FILIALES:
Centro: Esmeralda 320 - 3° (1343)
Capital Federal
Lanús: Caaguazú 2186
L. Este
Capital: Tucumán 2044 - 1° (1050) Capital Federal

Talent MSX
Inteligencia en crecimiento.

Centro para el desarrollo de la inteligencia.



Descubramos y construyamos juntos los caminos que nos permitirán el uso inteligente de los productos de la creatividad humana.

diálogo - *D.P.*

Titulos no oficiales

A la computadora personal

Talent MSX

nada le es imposible

dialogo - D.P.



MSX es marca registrada de MICROSOFT CORPORATION

Porque gracias a la norma internacional MSX, la TALENT MSX trasciende sus propios límites.
Hasta ahora, cuando usted compraba una computadora personal de cualquier marca, quedaba automáticamente desconectado del resto del mundo de la computación. Porque los distintos equipos y sistemas no eran compatibles entre sí.
Hasta que dos grandes empresas de informática, la Microsoft Corp. de EE.UU. y la ASCII del Japón se pusieron de acuerdo para crear una norma standard: la MSX. Que se expandió también rápidamente en Europa. Y que hoy TALENT presenta por primera vez en la Argentina.
Mientras que la mayoría de las computadoras de su tipo que se ofrecen en el mercado nacional, han sido discontinuadas por obsoletas en sus lugares de origen, TALENT MSX tiene casi ilimitadas posibilidades de desarrollo. Porque la norma MSX es en todo el mundo inteligencia en crecimiento.
La TALENT MSX pone a su disposición un mundo de software para elegir. Y con la incorporación de todos sus periféricos llega a ser una auténtica computadora profesional.

UTIL

Su poderoso sistema operativo MSX permite el acceso a todo tipo de procesamiento de datos:

- Planillas de calculo.
- Procesadores de palabra.
- Graficos de negocios.
- Bases de datos (d Base II, etc.)
- Contabilidad general, sueldos, y jornales, costos, etc., desarrollados bajo CP/M en Basic, Cobol, Pascal o C.

Con la posibilidad de conexión a linea telefonica permite la transferencia y consulta de datos entre computadoras personales, profesionales o bancos de datos.
La grabacion de archivos es en formato MS-DOS, haciendola compatible con las computadoras profesionales.

DIDACTICA

Dispone de tres lenguajes para la enseñanza de computacion: LOGO como lenguaje de induccion para los mas chicos. Lenguaje de Programacion en castellano, para todos los que quieran aprender a programar sin conocimientos previos. Y Basic MSX como lenguaje profesional. Mas una amplia variedad de perifericos como el Mouse, Lapiz Optico, Tableta grafica, Track-ball, etc.

DIVERTIDA

La mas genial para Video-Juegos. Por la amplisima biblioteca de programas **–todos nuevos–** de la norma MSX en el mundo. Y ademas, el Basic MSX permite al usuario generar sus propios juegos con un manejo tan simple, como solo TALENT MSX puede ofrecer.

CARACTERISTICAS TECNICAS

- Memoria principal 64 KB ampliable hasta 576 KB.
- Memoria de video: 16 KB RAM.
- ROM incorporada de 32 KB con el MSX-Basic de Microsoft.
- Graficos completos, hasta 32 sprites y 16 colores simultaneos.
- Generador de sonido de 3 voces y 8 octavas.
- Conexion para cualquier grabador.
- Interfaz para salida impresora paralela.
- Conectores para cartuchos y expansiones.
- Fuente para 220 V y modulador PAL-N incorporado.

DISTRIBUIDORES AUTORIZADOS: CAPITAL FEDERAL: AMATRIX, Bolívar 173 - ARGECINT, Av. de Mayo 1402 - BAIDAT COMPUTACION, Juramento 2349 - COMPUPRANDO, Av. de Mayo 965 - COMPUSHOP, Córdoba 1464 - COMPUTIQUE, Córdoba 1111, E. P. - COMPUTRONIC, Viamonte 2096 - CP67 CLUB, Florida 683, L. 18 - DALTON COMPUTACION, Cabildo 2283 - ELAB, Cabildo 730 - MICROSTAR, Callao 462 - Q.S.P., Bartolomé Mitre 864 - SERVICIOS EN INFORMATICA, Paraná 164 - DISTRIBUIDORA CONCALES, Tucumán 1458 - MICROMATICA, Av. Pueyrredón 1135 - **ACASSUSO:** MICROSTAR ACASSUSO, Eduardo Costa 892 - **AVELLANEDA:** ARGOS, Av. Mitre 1755 - **BOULOGNE:** COMPUTIQUE CARREFOUR, Bernardo de Irigoyen 2647 - **CASTELAR:** HOT BIT COMPUTACION, Carlos Casares 997 - **LANUS:** COMPUTACION LANUS, Caaguazú 2186 - **LOMAS DE ZAMORA:** ARGESIS COMPUTACION, Av. Meeks 269 - **MARTINEZ:** VIDEO BYTE, Hipólito Yrigoyen 32 - **RAMOS MEJIA:** MANIAC COMPUTACION, Rivadavia 13734 - **SAN ISIDRO:** FERNANDO CORATELLA, Cosme Beccar 249 - **VICENTE LOPEZ:** SERVICIOS EN INFORMATICA, Av. del Libertador 882 - **BAHIA BLANCA:** SERCOM, Donado 327 - SUMASUR, Alsina 236 - **LA PLATA:** CADEMA, Calle 7 Nº 1240 - CERO-UNO INFORMATICA, Calle 48 Nº 529 - **MAR DEL PLATA:** FAST, Catamarca 1755 - **NECOCHEA:** CAFAL, Calle 57 Nº 2920 - SERCOM, Calle 57 Nº 2216 - **TRENQUE LAUQUEN:** COMPUQUEN, Villegas 231 - **CORDOBA:** AUTODATA, Pasaje Santa Catalina 27 - TECSIEM, Santa Rosa 715 - **ROSARIO:** 2001 COMPUTACION, Santa Fe 1468 - MINICOMP, Maipú 862 - SISOR, Urquiza 1062 - **SANTA FE:** ARGECINT, P. San Martin 2433, L. 36 - SISOR, Rivadavia 2553 - INFORMATICA, San Gerónimo 2721/25 - **VILLA MARIA:** JUAN CARLOS TRENTO, 9 de Julio 80 - **MENDOZA:** INTERFACE, Sarmiento 98 - BIT & BYTE, 9 de Julio 1030 - **COMODORO RIVADAVIA:** COMPUSER, 25 de Mayo 827 - **GENERAL ROCA:** DISTRIBUIDORA VECCHI, 25 de Mayo 762 - **LA PAMPA:** MARINELLI, Pellegrini 155 - **NEUQUEN:** MEGA, Perito Moreno 383 - EDISA, Roca esq. Fotheringham - **RIO GRANDE:** INFORMATICA M & B, Perito Moreno 290 - **SAN CARLOS DE BARILOCHE:** L. ROBLEDO & ASOCIADOS, Elfein 13, Piso 1º - **TRELEW:** SISTENOVA, Sarmiento 456 - **PARANA:** MARIO GARCIA, Laprida y Santa Fe - **POSADAS:** CENTRO DE COMPUTOS ELDORADO, Colón 2429 - **RESISTENCIA:** FRANCO SANTI, Carlos Pellegrini 761 - **SAN SALVADOR DE JUJUY:** DELTA COMPUTACION, Salta 1108 - **SALTA:** DELTA COMPUTACION, Caseros 873 - **SAN MIGUEL DE TUCUMAN:** LEXICON, 9 de Julio 85 - ELECTRONICA